

**** IcePRO 3.0 ****

User's manual

I.	What is IcePRO ?	4
a)	Why ?	4
b)	What ?	4
c)	How ?	4
d)	Compatibility	6
II.	Configuration	7
a)	Sites	7
b)	Databases	7
III.	Getting started	8
a)	Applications, versions, environments	8
b)	Root directory	9
c)	Create your first environment	10
d)	Using your environment	12
e)	Databases and connections	13
f)	Register sources	14
g)	And now ?	16
IV.	Managing applications	16
a)	Sources	16
b)	Modules	17
c)	Versions	17
d)	Development environment	21
e)	Sources exchange between sites	22
f)	Application deployment	24
V.	IcePRO-ADE interactions	28
a)	ADE Tools	28
b)	IcePRO Compiler	29
c)	“Compile on save” feature	31
d)	Saving files	31
e)	Moving or renaming files	32
f)	Automatic file backup	33
g)	First level history records	33
h)	Sources screen	34
i)	Consistency control	34
VI.	Evaluation	35
a)	Source analysis	35
b)	Uses	37
c)	Unused objects	40
d)	Call hierarchy	41
e)	Field search	42
f)	Version statistics	44
g)	Session analysis	44
VII.	Creation	48
a)	Skeleton	48
b)	Constants	50
c)	Source generation	51
d)	Functions libraries	51
VIII.	User interface	52
a)	Screens	52
b)	Exporting to Excel	55

c) Creating buttons	56
d) Login	57

I. What is IcePRO ?

a) Why ?

There is no doubt that Progress development environment is one of the best you can find on the market. Powerful language, strong and efficient database and so on. If you use it, you like it. But it is not enough to produce high quality applications and maintain them at low cost. IcePRO helps you to develop, to manage, to evaluate and to deploy your applications.

b) What ?

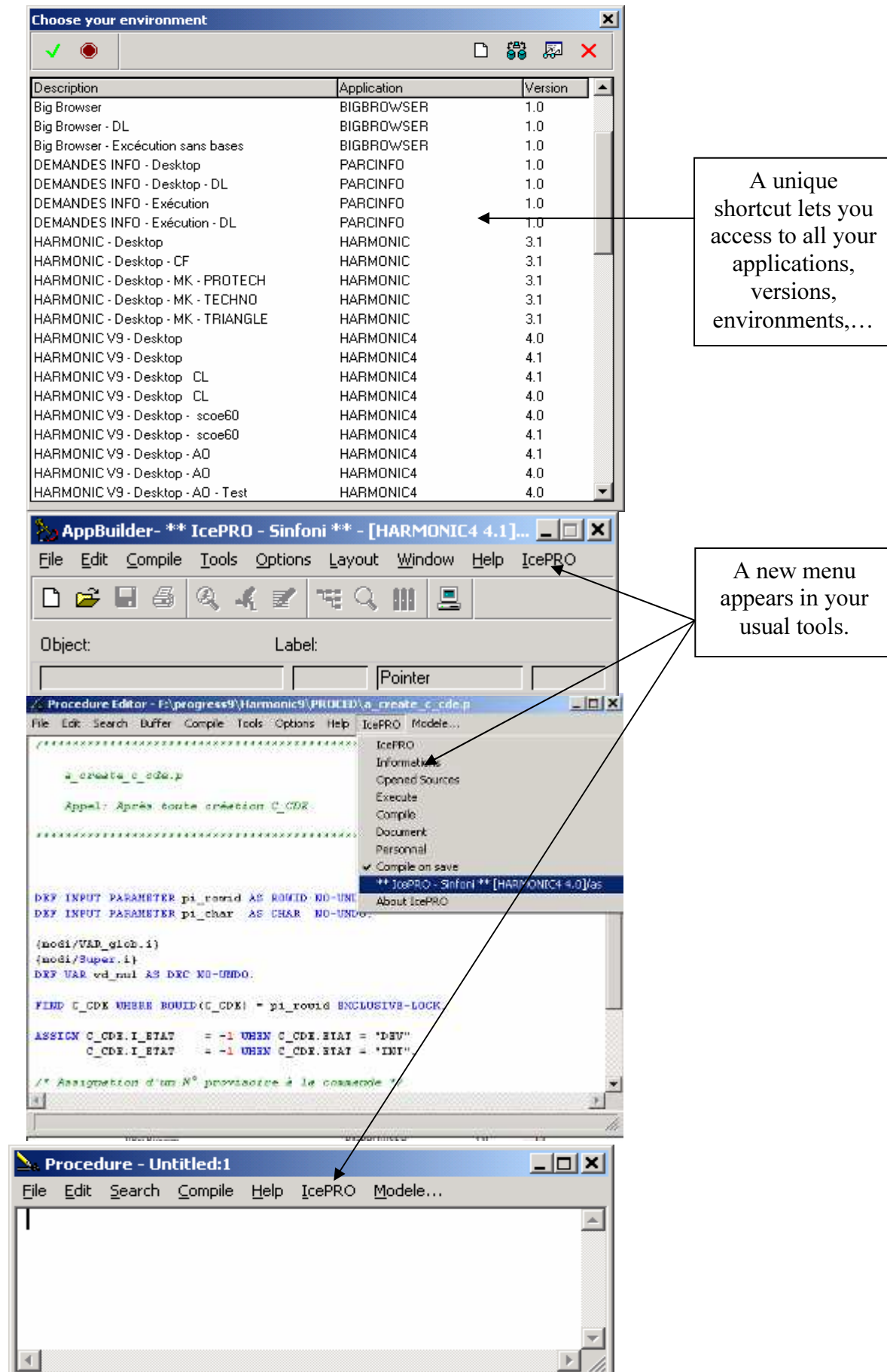
IcePRO is a software workshop created to help Progress developer teams to manage Progress applications. It is divided in three parts:

- ⇒ **I**ntervention: Manage source code, applications versions, multi sites configurations. Deploy updates. Keep history of modifications, document them.
- ⇒ **C**reation: Source code generator.
- ⇒ **E**valuation: Analyze your sources code and queries, analyze your applications while running (profiler).

c) How ?

IcePRO is fully developed in Progress 4GL. Thus, it is integrated to your usual Progress environment. By installing IcePRO, you get one shortcut on your desktop to work with all your applications, in different configurations if needed (production, tests,...). Using this shortcut, a new menu appears in Progress ADE tools (AppBuilder, Procedure Editor, procedure windows).

IcePRO comes with a database. This database is not needed for the deployment. There is no part of IcePRO you need to deploy with your application.



d) Compatibility

IcePROV930setup.exe : Progress 9.1, undefined code page.

IcePROV9ISO30setup.exe: Progress 9.1, ISO8859-1 code page.

IcePROV1030setup.exe: Progress10.1, undefined code page.

IcePROV10ISO30setup.exe: Progress10.1, ISO8859-1 code page.

According to Progress r-code compatibility, you may run

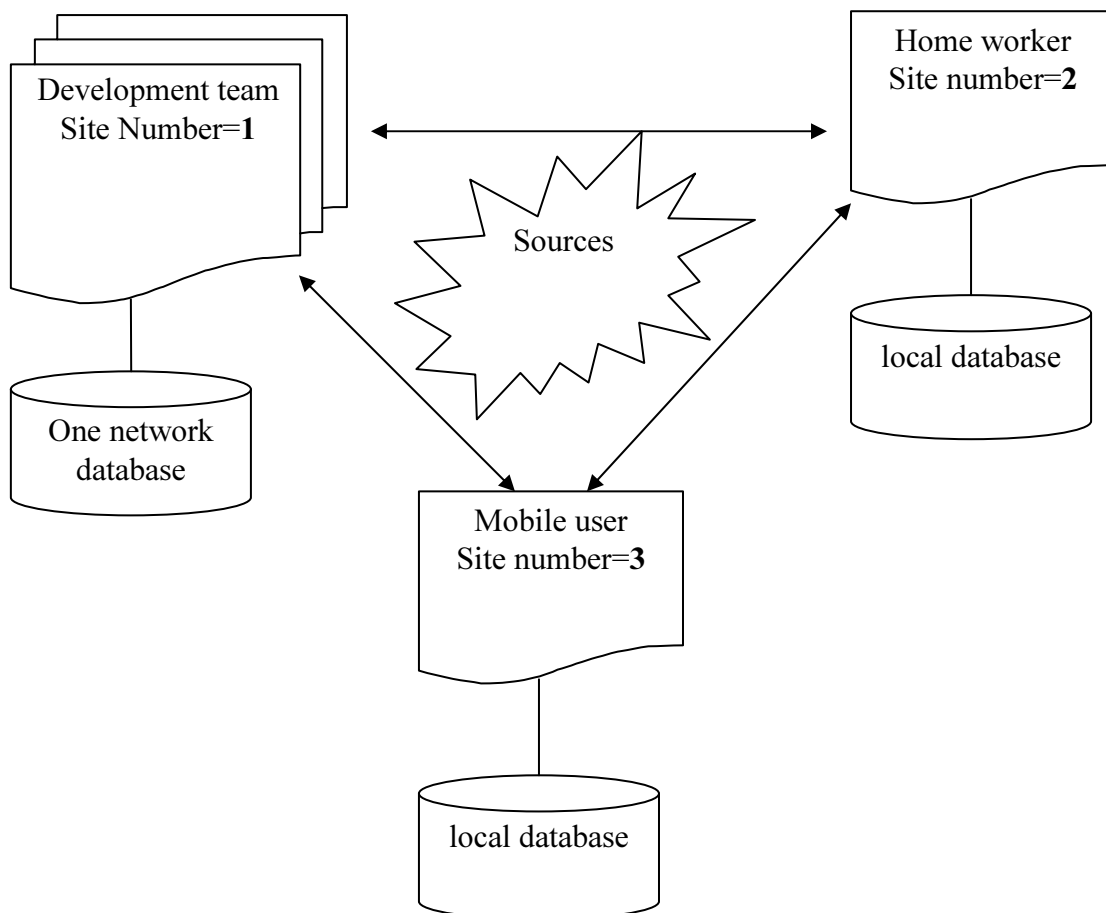
IcePROV1030setup.exe / IcePROV10ISO30setup.exe with 10.X releases
and

IcePROV9ISO30setup.exe / IcePROV9ISO30setup.exe with 9.X releases
without any warranty.

II. Configuration

a) Sites

IcePRO is able to work in a multi sites environment. You can have several development teams, home workers, mobile users. Each site works with its own IcePRO database and must have a unique site number. When needed, each site can send/receive updates to/from other sites to synchronize modified sources and their histories. Each source registered on a site receives a unique ID containing the site number. So, you can not exchange sources between sites with the same site number. When you install IcePRO, a default site named DEMO is created with number 1. If you install IcePRO on more than one site in order to synchronize sources, you have to change the name and number for each site.



b) Databases

IcePRO is provided with V9 or V10 database and its definition file (df). In the default configuration, this database is installed as a local database in the IcePRO installation directory with the file IcePRO.pf (parameter file) which is used for the connection.

For a developer team, use this database or the df file to create a network database and modify IcePRO.pf according to the new database connection parameters (-S, -H,...).

For home workers or mobile users, you can keep the default local database. If they need to run several instances of IcePRO at the same time (for instance, a development session and a

runtime session), you need to run a local server for the database (by using proserve). In this case, just remove the “-l” parameter in IcePRO.pf file.

Note : If you have to reinstall IcePRO on a computer, keep in mind that the installation process can override an existing local database and parameter file. If you uninstall IcePRO, the local database and parameter file can be deleted. Unless you use a network database, be sure to have a backup of your local database before proceeding. Even if you use a network database, IcePRO uses the local IcePRO.pf file to connect it. Always make a backup of this file before proceeding.

All IcePRO tables names have a trailing “_” character (appli_, version_, ...) to avoid conflict with other connected databases. All IcePRO files are located in “al” subdirectory. Do not use such subdirectories with your application to avoid conflict.

III. Getting started

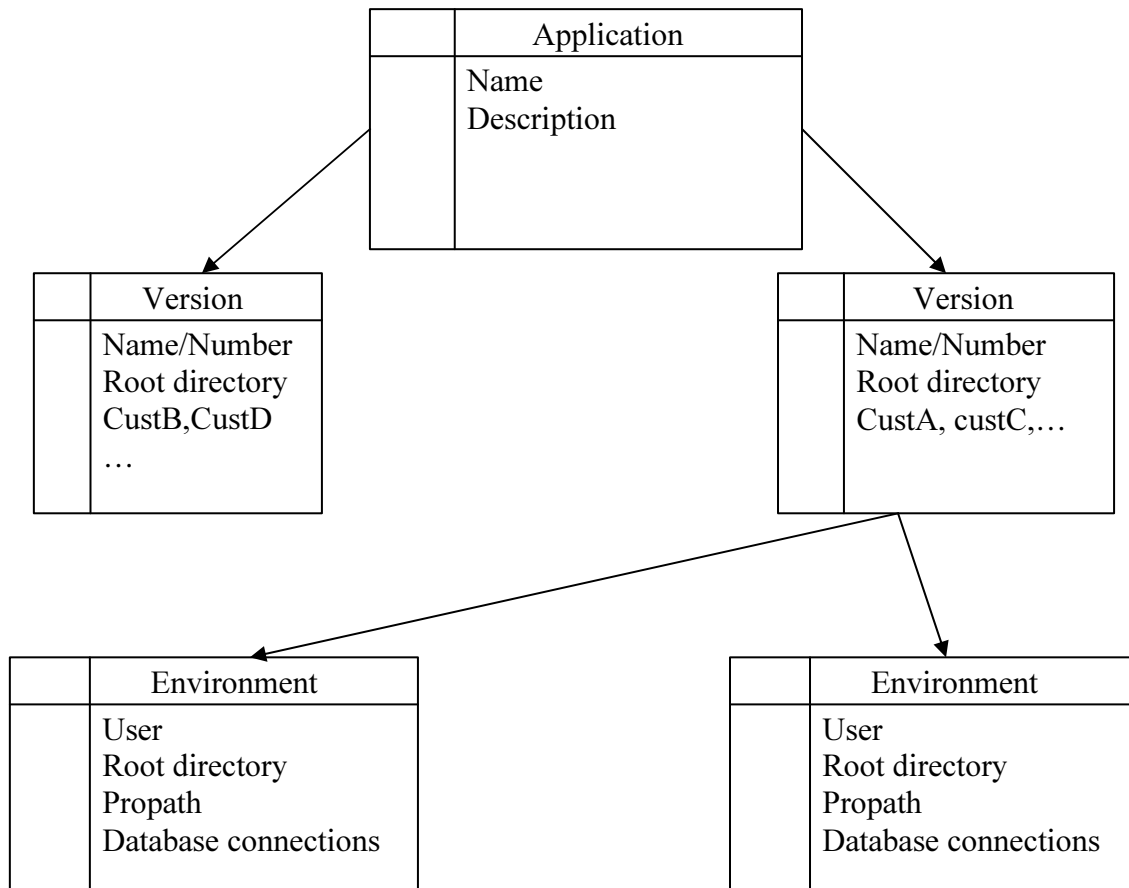
a) Applications, versions, environments

These are the essential concepts. Application is the top level management unit. For each application, you may create several versions. For each version, you may create several environments.

An application is defined by a **name** and a description.

A version is defined by the **application** to which it belongs, a **name** (or number: 1.0, 2.1,...), a description, a **root directory**, and the list of the customers who use it.

An environment is defined by the **user** to who it belongs, the **application** and **version** for which it was created, the **root directory** (can be different from the version root directory, but points to it), the **propath** value, and the **database connection parameters**.

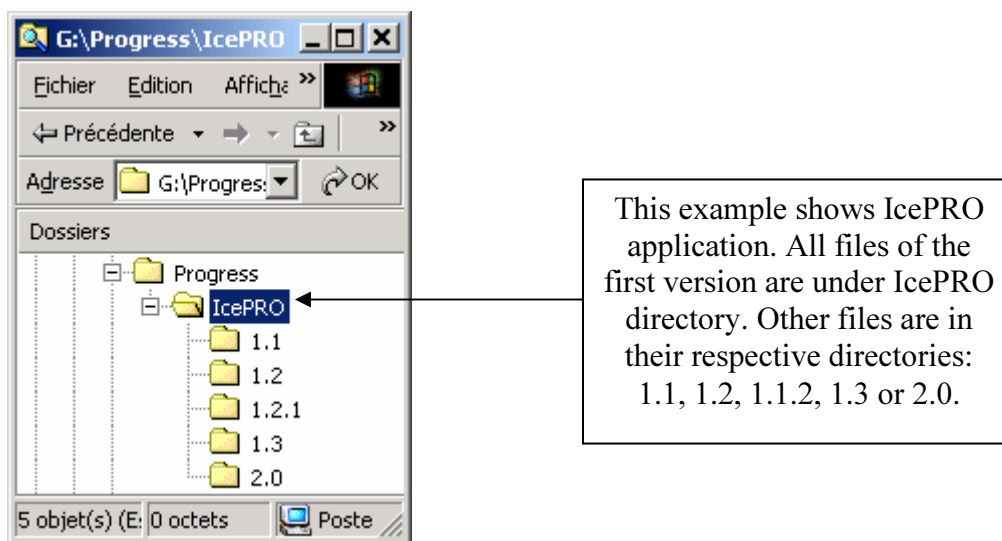


b) Root directory

The version **root directory** is the physical local or network directory where all application files can be found. IcePRO considers that a file belongs to a version only if it can find it in this directory or subdirectories.

The UNC (that is `\\hostname\path`) must be used for the version **root directory** in a network environment. However, if you use IcePRO in a stand alone database configuration, UNC is not required.

You can choose only the first application version **root directory**. When you add a version, its **root directory** is automatically defined (and created) as a subdirectory of the first version. The subdirectory name is the version name/number:



IcePRO automatically reproduces the subdirectories structure of the version n in the directory of the version $n+1$.

c) Create your first environment

After you have created the first user by using Progress Data Administration (IcePRO runs it automatically for you), restart IcePRO. The first screen lets you create an application and a version. It can be an existing application:

The 'Create a version' dialog box contains the following fields and annotations:

- Application:** MyFirstAppli (An arrow points from this field to a box labeled 'Application name and description').
- Version:** 1.0 (An arrow points from this field to a box labeled 'Version name/number').
- Location:** G:\Progress\IcePRO (An arrow points from this field to a box labeled 'Version root directory. Use UNC if the directory is on a network..').
- Location (UNC Name):** G:\Progress\IcePRO
- Documentation directory:** (Empty field)
- Development directory:** (Empty field)
- Description:** My first application (An arrow points from this field to the 'Application name and description' box).

Documentation and development directories are optional and can be typed later. The documentation directory receives Word documents about sources generated by IcePRO. The development directory is used in development process.

Valid the form, and choose the version you just have created. (Note that at this point, you can add more couples of applications/versions by using “New...” button, but you can not add a version to an existing application by this way.)

The 'Choose a version' dialog box displays a list of created versions. The list has two columns: 'Application' and 'Version'. The entry 'MyFirstAppli' with version '1.0' is selected and highlighted. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'New...'.

Application	Version
MyFirstAppli	1.0

User environ

✓ ⚠

User: as

Application: MyFirstAppli

Version: 1.0

Root (UNC Name): G:\Progress\IcePRO

Description:

Root (User acces):

Startup procedure:

☐ Run Desktop

1. [root]
2. [current directory]
3. f:\dlc91\gui
4. f:\dlc91\gui\adecomm.pl
5. f:\dlc91\gui\adecomp.pl
6. f:\dlc91\gui\adedesk.pl
7. f:\dlc91\gui\adedict.pl
8. f:\dlc91\gui\adededit.pl
9. f:\dlc91\gui\adeicon.pl
10. f:\dlc91\gui\aderes.pl
11. f:\dlc91\gui\adeshar.pl
12. f:\dlc91\gui\adeuib.pl
13. f:\dlc91\gui\adeweb.pl

Checked: IcePRO runs Progress Desktop.
Unchecked: IcePRO runs the startup procedure you have defined.

Generally, that is the procedure you use to run your application. It is used if you don't check the box "Run Desktop", so IcePRO runs your application instead of Progress desktop. It is also used to run real time session analysis.

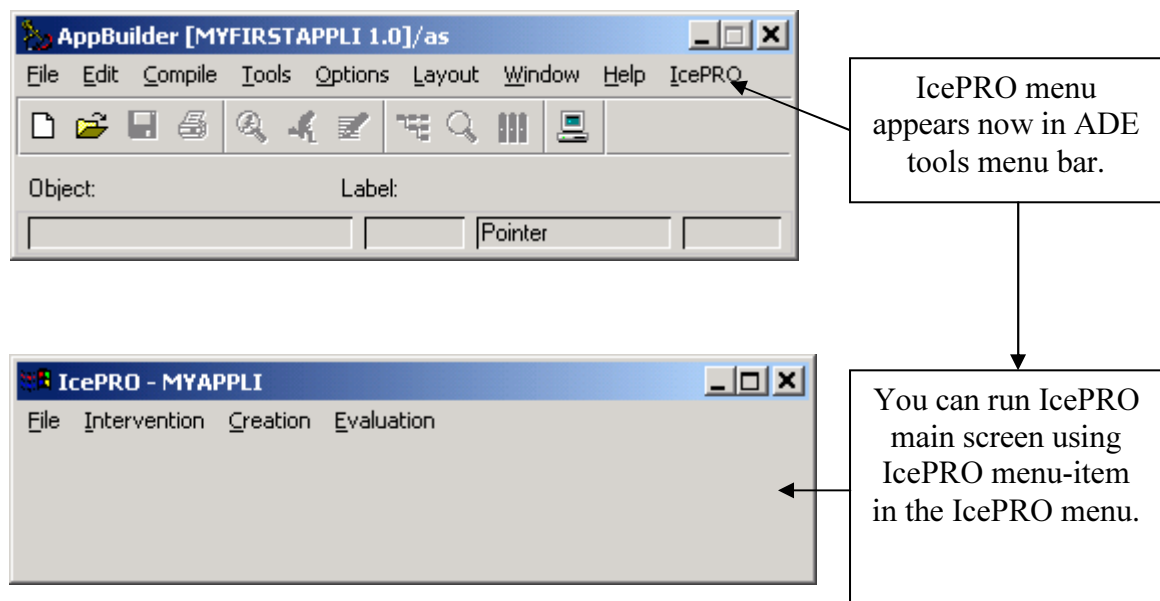
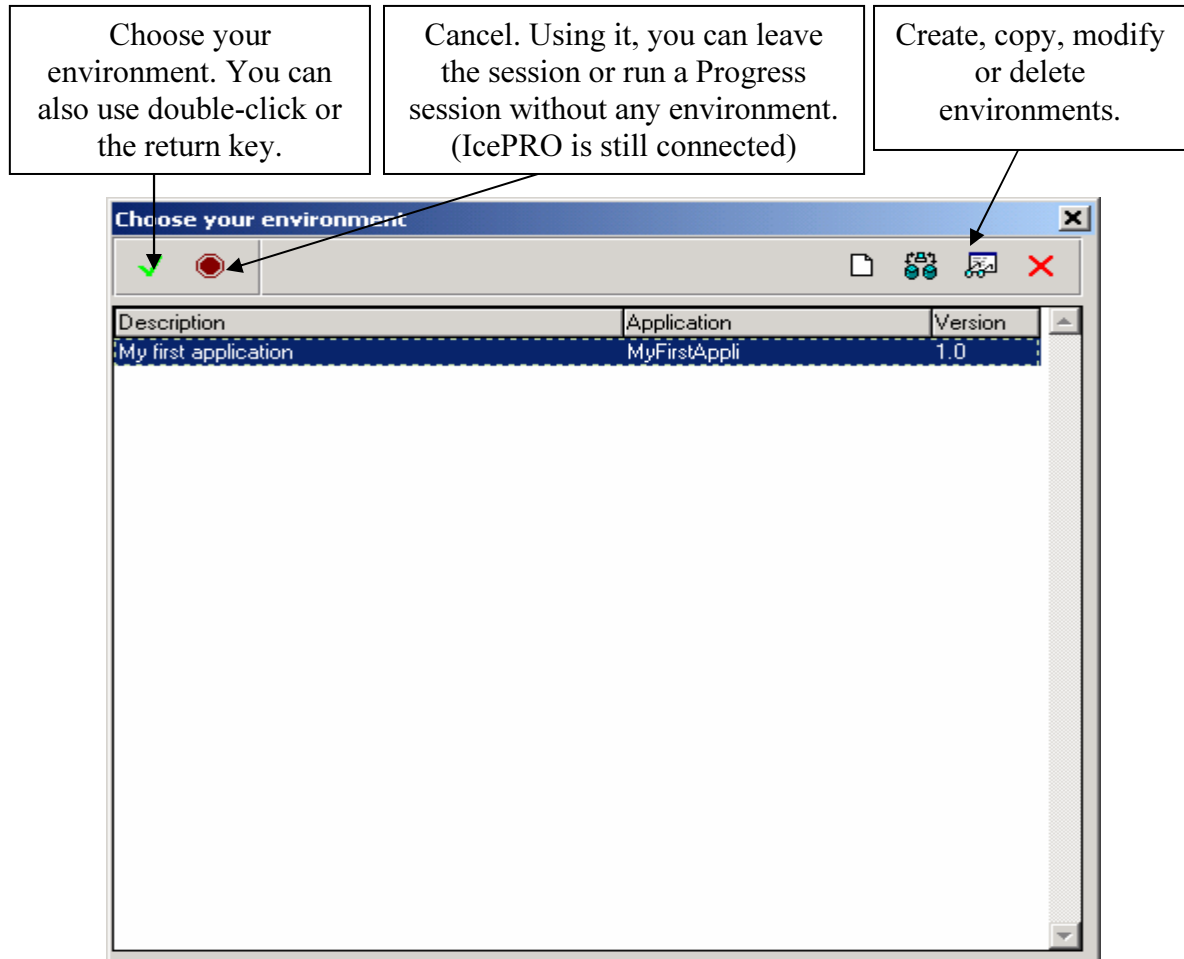
This **root directory** means how **you** access to the files. Use a network drive, not UNC. The path can be different from the version root because you use a network drive different from other users. However, the conversion in UNC must match the version root directory.

This the environment proppath. [root] is an alias to the root directory. You add path using it:[root]\<your path>. Never use full pathname.

Use this editor to type your database(s) connection parameters:
-db mybase1 -S mysrv1 -H myhost1
-db mybase2...
You can not use session startup parameters here (-d dmy, -s,...) because the progress session is already running.

d) Using your environment

Once you have finished to provide the informations about you application, the following screen appears. It will be the entry point for all the applications you manage with IcePRO. As each environment belongs to a user, you can only see the environments you have created.



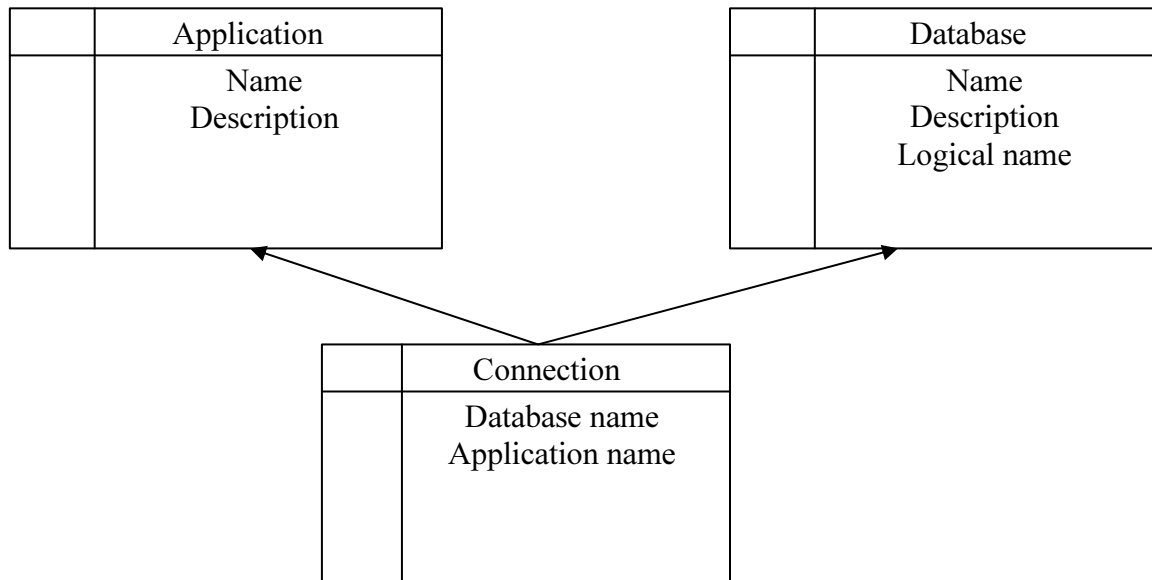
e) Databases and connections

IcePRO database contains a DATABASE table and a CONNECTION table. From IcePRO main screen, you can manage:

Applications by using the menu File->General->Applications

Databases by using the menu File->General->Databases

Connections by using the menu File->Application->Connections



It allows you to describe the relationship between applications and databases. IcePRO needs this information for analysis, code generation and compilation purpose. Be sure to keep consistence between the real database connections (defined in your environments) and the connection table. When you run a session, IcePRO verifies that all needed databases are connected. A warning is displayed if one or more databases are not connected.

Example:

Application: MyAppli

Databases: MyBase1, MyBase2 (these are logical name)

Connections: (MyAppli,MyBase1), (MyAppli,MyBase2).

For each environment using MyAppli, you must provide “-db MyBase1 -db MyBase2” connection parameters.

Note : You can use every name you want in the database table. But the logical name must match your connection parameters. That is, you can create a database record where Name=“DBName” and Logical Name=“DBLogName”, then create an environment which connects to a physical database named “DBPhys”. In this case, you must provide “-db DbTest -ld DBLogName” (see Progress help for more information about -ld parameter).

Testing a WebClient application is a case where you do not provide any database connection parameter. IcePRO displays a warning. Just ignore it.

Do not create database or connection which refers to the IcePRO database.

f) Register sources

A source is a file needed by your application, not only 4LG. These files are defined by their extensions. In the default configuration, IcePRO provides this types:

Source type	Description	Extension
BMP	Image files	bmp
ICO	Image files	ico
INC	Includes	i
OCX	OCX files	ico
PRL	Report Builder library	prl
PROC	Procedure	p
UDF	User defined function (RB)	udf
WIN	Screens	w
WRX	WRX files	wrx

To add, remove or edit source types, use menu *File->General->Sources types*. For instance, you may want to manage batch files, ini files, or even exe files.

For each type, you choose if the files are compilable, and if the file date must be used to detect modifications.

Compilable files : All file which can be compile in .r file. For instance, .p file must be compiled, includes have not to be compiled.

Use file date: As IcePRO is integrated with ADE Tools, it detects all modified files edited in Appbuilder, Procedure Editor and Procedure Window. There is no need to use file date to detect them. On the contrary, for the files modified with other tools (e. g. PRL files, .exe, ...), IcePRO must use file date in order to determine if a file was modified.

Note : Do not create a type for the ".r" extension.

Register a file means IcePRO records it in its database. There are two conditions for a file to be registered: It must be in the version root directory and its extension must be defined in the source types.

There are several ways to register a file:

- ⇒ When you save it for the first time with ADE tools.
- ⇒ While application analysis.
- ⇒ When you run the IcePRO search file function.

.

.

The last case is useful if you start to manage an existing application. Run the Sources management program *File->Application->Sources*.

Search sources

Sources [BASE/as] - MyFirstAppli affaire bmp BMP PROCED\al\image

File

☐ Display only thin client (server and client side)
☐ Display absents

Version	Source	Extension	Kind of source	Module	Modified	Modified by	Location
1.0	affaire	bmp	BMP		06/06/2002	as	PROCED\al\image
1.0	avance	bmp	BMP		29/08/2001	as	PROCED\al\image
1.0	barcode	bmp	BMP		08/12/1999	as	PROCED\al\image
1.0	BL	BMP	BMP		16/04/2002	as	PROCED\al\image
1.0	calc	bmp	BMP		22/04/2002	as	PROCED\al\image
1.0	calendar	bmp	BMP		10/12/2001	as	PROCED\al\image
1.0	CDE	BMP	BMP		16/04/2002	as	PROCED\al\image
1.0	CHECK	BMP	BMP		21/11/1998	as	PROCED\al\Adeiconpl
1.0	CHECK	BMP	BMP		21/11/1998	as	PROCED\al\image
1.0	check	p	PROC		06/08/2002	as	PROCED\al\ske
1.0	clear	bmp	BMP		01/03/2000	as	PROCED\al\image
1.0	client	bmp	BMP		21/01/2002	as	PROCED\al\image
1.0	Cloturer	BMP	BMP		29/08/2001	as	PROCED\al\image
1.0	CNFGINFO	BMP	BMP		21/11/1998	as	PROCED\al\Adeiconpl
1.0	CNTRLHRY	BMP	BMP		21/11/1998	as	PROCED\al\Adeiconpl
1.0	corbeille	bmp	BMP		21/09/1999	as	PROCED\al\image
1.0	CROSS	BMP	BMP		21/11/1998	as	PROCED\al\Adeiconpl
1.0	CROSS	BMP	BMP		21/11/1998	as	PROCED\al\image
1.0	CUEEXIT	BMP	BMP		21/11/1998	as	PROCED\al\Adeiconpl
1.0	CUEEXIT	BMP	BMP		21/11/1998	as	PROCED\al\image
1.0	CUEHELP	BMP	BMP		21/11/1998	as	PROCED\al\Adeiconpl
1.0	cuehelp	bmp	BMP		21/11/1998	as	PROCED\al\image
1.0	date	bmp	BMP		10/12/2001	as	PROCED\al\image

Search sources...

☒ ☐

Start in: G:\Progress\IcePRO\

Excluded directories:

+

-

Launch the search.

Add/Remove excluded directories.

g) And now ?

You can work as you usually do with the ADE Tools and benefit from IcePRO features. In the next pages of this manual, you will find two kinds of menu references. The references beginning with ADE->IcePRO refer to the IcePRO menus added to Appbuilder, Procedure Editor and Procedure Windows (ADE->IcePRO->IcePRO runs IcePRO main screen, ADE->IcePRO->Compile runs IcePRO application compiler). Other references refer to IcePRO main screen menu (File->General->Applications runs Application management screen).

IV. Managing applications

a) Sources

File->Application->Sources

A source (or a file) is defined by:

- ⇒ Application : The application to which it belongs.
- ⇒ Version : The application version
- ⇒ Location : Relative path to the version root directory.
- ⇒ Name : File name without path nor extension.
- ⇒ Extension : File name extension (without the dot).

Note : A source has also an internal number which is unique through all sites.

Other informations:

- ⇒ Description : This allows you to shortly describe the source in IcePRO database.
- ⇒ Module : Module application to which the file belongs.
- ⇒ Type : Defined by its extension.
- ⇒ Compile on save : This purpose allows you to automatically compile 4LG when you save it in the Procedure Editor or a Procedure Window (even .p files).
- ⇒ History : Tells IcePRO to track the source modifications and manipulations.
- ⇒ Thin client – Server Side/Client Side : Helps you to generate WebClient package.
- ⇒ Access: Access are defined at module or source level. Access work by using Progress CAN-DO syntax. If a user has no right on module or source, it can not save the source from ADE tools, nor modify data in IcePRO source screen.
- ⇒ Modified/Created/Analyzed: Date, hour and user. Note that “Creation” means creation in IcePRO database. For file created with ADE while IcePRO was running, this is also the user who has created the file.
- ⇒ The folder “General” also has a free text zone where you type any information.
- ⇒ History folder : Shows source history, that is : What ? (modification, creation, analysis, send, receive, ...), Who ? When ? With which product (IcePRO, EDT=Procedure Editor, UIB=AppBuilder or PRC=Procedure Window). When you modify a source by using ADE, IcePRO ask you to describe what you have done (One record per day).
- ⇒ Table, Index and Includes/Calls folders are useful informations collected by IcePRO when you run an analysis.

Note that a source is not necessarily a Progress 4GL file. It may be any file located under the version root directory with an extension defined in the list of source types. Typically, you define source types in a manner that all files required to deploy your application are registered.

Once registered, you can perform the following actions with a source:

- ⇒ Document: Create a Word document to type any relevant information. Documents are saved in the version document directory.
- ⇒ Static and dynamic analysis (4GL sources only): Which tables, fields, index are used, inclusion tree, call tree,...
- ⇒ Track modifications (ADE tools only) : You can describe all modifications you have made when you close a source after having saved it.
- ⇒ Rename or move file (ADE tools only) : When you use “Save as...” in AppBuilder, Procedure Editor or Procedure Window, IcePRO can delete original file. This prevents you to leave unused files in your application directories.
- ⇒ Synchronize source between multiple sites.
- ⇒ Put a source in a Development environment : A copy of the file is created in a particular directory where you can modify it.
- ⇒ Put a source to a higher version : A copy of the file is created in a higher version root directory.
- ⇒ Search r-code Database references : Useful when working on WebClient applications to check that client side programs does not make direct database access.
- ⇒ Compile it : The target directory depends on the version you are managing.
- ⇒ Create deployment packages, upgrade packages.
- ⇒ Create XRef, listing, and debug files for debugging purpose.
- ⇒ Search a particular source by using the filters available in the source program management.

b) Modules

File->Application->Module

Module is optional but useful. With modules, you can:

- ⇒ Quickly search files by using this criteria.
- ⇒ Make deployment packages by using this criteria.
- ⇒ Define access rights : Unless it has its own rights defined, a file inherits from the rights of the module to which it belongs. Use the field “Access” to define access rights.
- ⇒ Isolate customers specific sources.

c) Versions

File->Application->Versions

IcePRO allows you to manage versions. What does it mean ?

- ⇒ A same file can exist in different versions.
- ⇒ You are able to compile and/or run any version at any time by using your IcePRO environments.
- ⇒ You can maintain all versions of an application.

How does it work ?

IcePRO uses the progress PROPATH. When you create the first version of an application, you provide the version root directory. This root directory can contain subdirectories. This subdirectories tree depends on the way you want to organize your applications file. Imagine you create an application named "MyAppli", and its first version called "1.0."

Note : To create an application, use File->General->Application. Click "New" button, type the application code, valid the form a first time, then type its description and valid again. When you create a record in IcePRO database, the process is always in two phases. First, you type fields corresponding to the primary key and you valid the form a first time. Then you can provide other informations.

Once you have created your application, you can create the first version by using File->Application->Versions.

Note: You can double click on blue fields to choose among available records. When you type the version root directory, IcePRO creates it if it does not exist.

Suppose you have chosen **E:\MyAppli** as the version root directory and you have created the following subdirectories:

Proc: contains Progress .p files

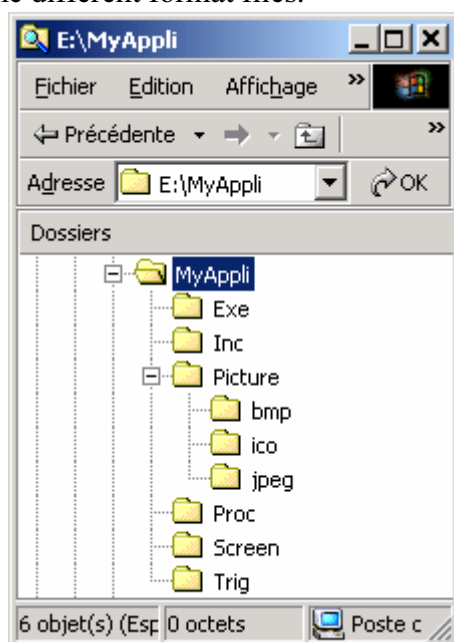
Inc: Contains Progress .i files.

Screen: Contains Progress .w files.

Trig: Contains triggers.

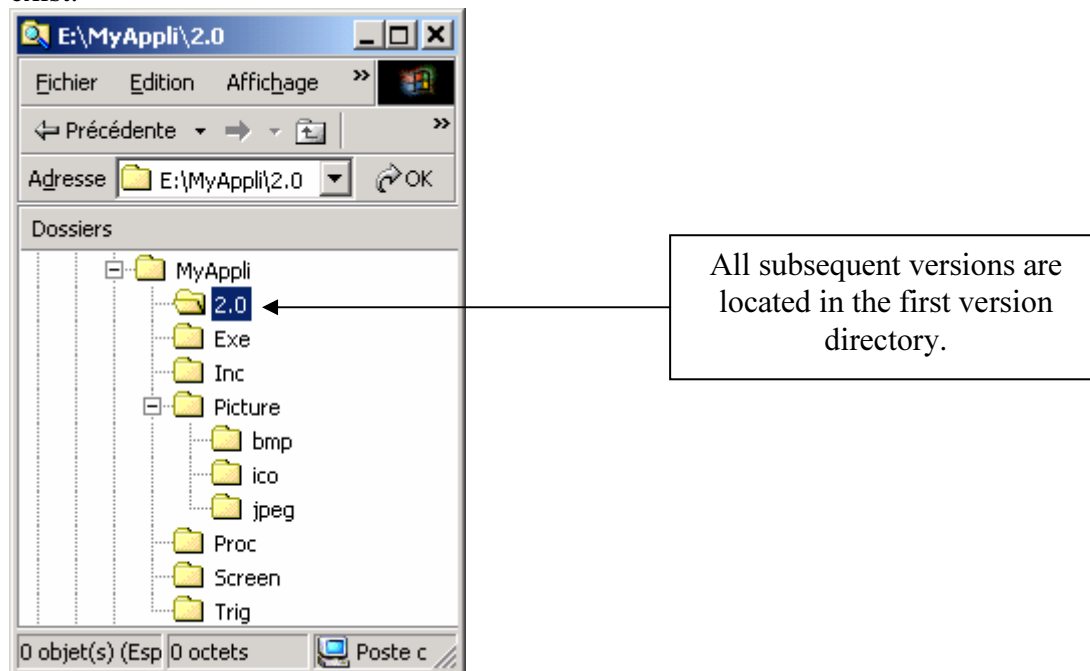
Exe: Contains some executable files or DLL.

Picture: contains three subdirectories named bmp, ico and jpeg. These subdirectories contain the different format files.



When you create your environment, remember that the alias [Root] represents *E:\MyAppli* directory. Suppose now that your startup procedure is LOGIN.W located in “screen” subdirectory, you have to add at least *[root]\Screen* to you propath. Then, you can choose to reference any other files by using subdirectory in your programs, that is *{inc\myinclude.i}, run proc\myproc.p, LOAD-IMAGE(“Picture\bmp\Mypicture.bmp”), ...* or not. On the one hand, you have finished with the propath, on the other hand, you also have to add *[root]\proc, [root]\inc, ...* and so on. It is your choice.

You can now work with the first version, and create hundreds of files in the tree. You may have several customers using version 1.0 and you want to create a new version with some modifications. To do that, you run *File->Application->Versions* to create the 2.0 version. First of all, you have not to choose the version root directory because IcePRO creates it automatically for you. It is *E:\MyAppli\2.0*. This subdirectory is also created if it does not exist.



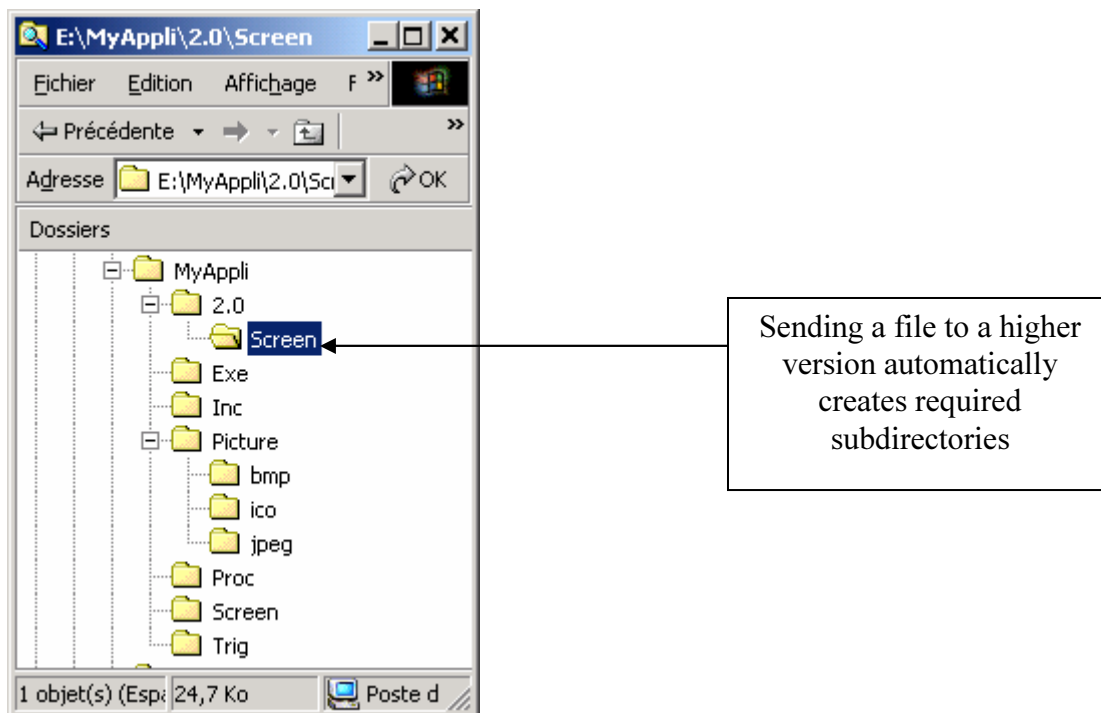
Then, the environment which refers to the 1.0 version has been duplicated. The copy refers to the 2.0 version, with the same propath, but the [root] alias has a different value because the V2.0 root directory is *E:\MyAppli\2.0*.

As it is not the goal to copy all your files from 1.0 to 2.0, there is also a “hidden” part of propath. Indeed, for each propath entry containing the [root] alias, other entries are created by IcePRO to previous versions. So, if you see *[root]\Screen* in version 2.0, the real propath is *E:\MyAppli\2.0\Screen, E:\MyAppli\Screen*. If you have a third version, *[root]\Screen* will be equivalent to *E:\MyAppli\3.0\Screen, E:\MyAppli\2.0\Screen, E:\MyAppli\Screen* and so on.

This mechanism allows you to run your application under Vn version immediately, because if Progress session can not find a file in *E:\MyAppli\N*, it will find it automatically in *E:\MyAppli\N-1* or *E:\MyAppli\N-2, ...* It means that you only have to move files which have to be modified from a version to another. To do this, use *File->Application->Sources*.



IcePRO creates the required subdirectories on the fly, and then copy the file. The source now exists for the two versions, and there are two records in IcePRO, one for each version.



Note: When a file exists in version n , it is not a very good idea to modify it in the version $n-1$ because there is a high risk of inconsistency between the versions. So, if IcePRO lets you do it, it displays a warning when you open such a file in ADE tools. You always should to duplicate all modifications you have made in a version to all subsequent versions.

What about includes and databases modifications ?

Often, moving from a version to another involves deep modifications, as database schema or include files. This can affect the whole application. For instance, modifying the database

schema can prevent you to run r-code from previous version because of CRC. In this case, you just have to compile your application by using IcePRO compiler

ADE->IcePRO->Compile

Note: You must have defined the databases and connections

(File->General->Database and File->Application->connections) before you can use IcePRO compiler.

The IcePRO compiler put the r-code files in the right directory according the environment you are currently running. It creates all required subdirectories. Compiling files under environment which refers to 2.0 version will create all subdirectories which does not exist in **E:\MyAppli\2.0** directory. If a file does not exist in version 2.0, the 1.0 version will be compiled in its place and the r-code will be generated in **E:\MyAppli\2.0**. When compiling in 2.0, all existing includes in 2.0 version are used. This allows you to always be able to run one of the available version of your application.

d) Development environment

When you create a version, you can fill a field named “Development directory”. This directory must be a subdirectory of the version root directory. This development directory can be used to put files in it before modification. If you put a file in development environment, IcePRO copies this file in the development directory. If this directory is **E:\MyAppli\2.0\Dev** and you put **E:\MyAppli\2.0\Screen\login.w**, you will get this file:



E:\MyAppli\2.0\DEV\Screen\login.w.

To test your modification while you are modifying your source, create a development environment where you define a propath containing this directory before all others: If the default version path is **[root]\Screen**, create an environment with this propath: **[root]\Dev\Screen, [root]\Screen**. This way, this environment will run the modified source instead of the original version.

There are several advantages to use development environment:

- ⇒ All others developers are alerted when they try to open the original source.
- ⇒ You keep your original file available.
- ⇒ Your modification does not affect other environments/developers.

Note: To put/get a file in/from development environment run

File->Application->Sources, and use /  buttons.

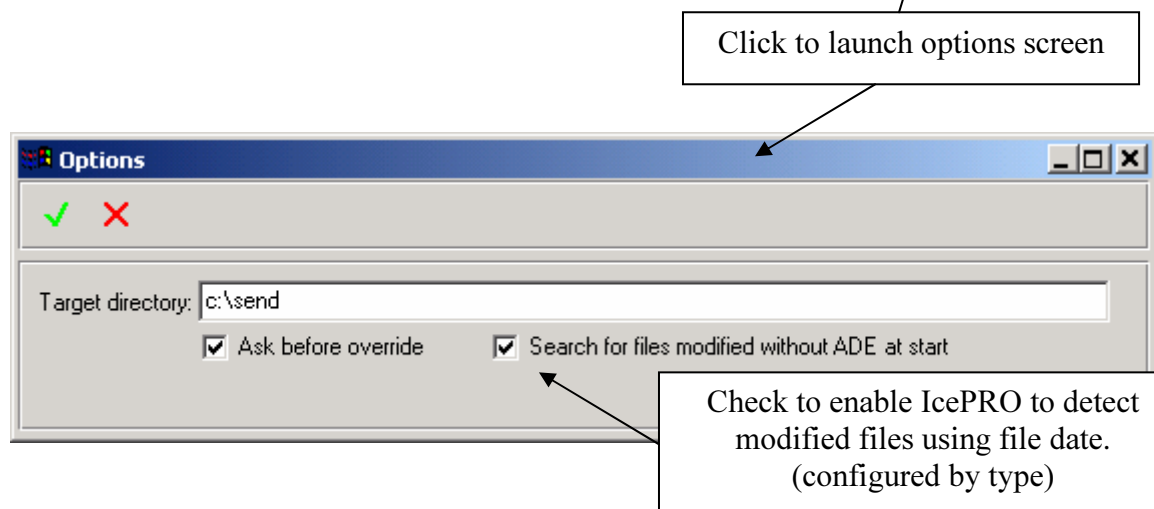
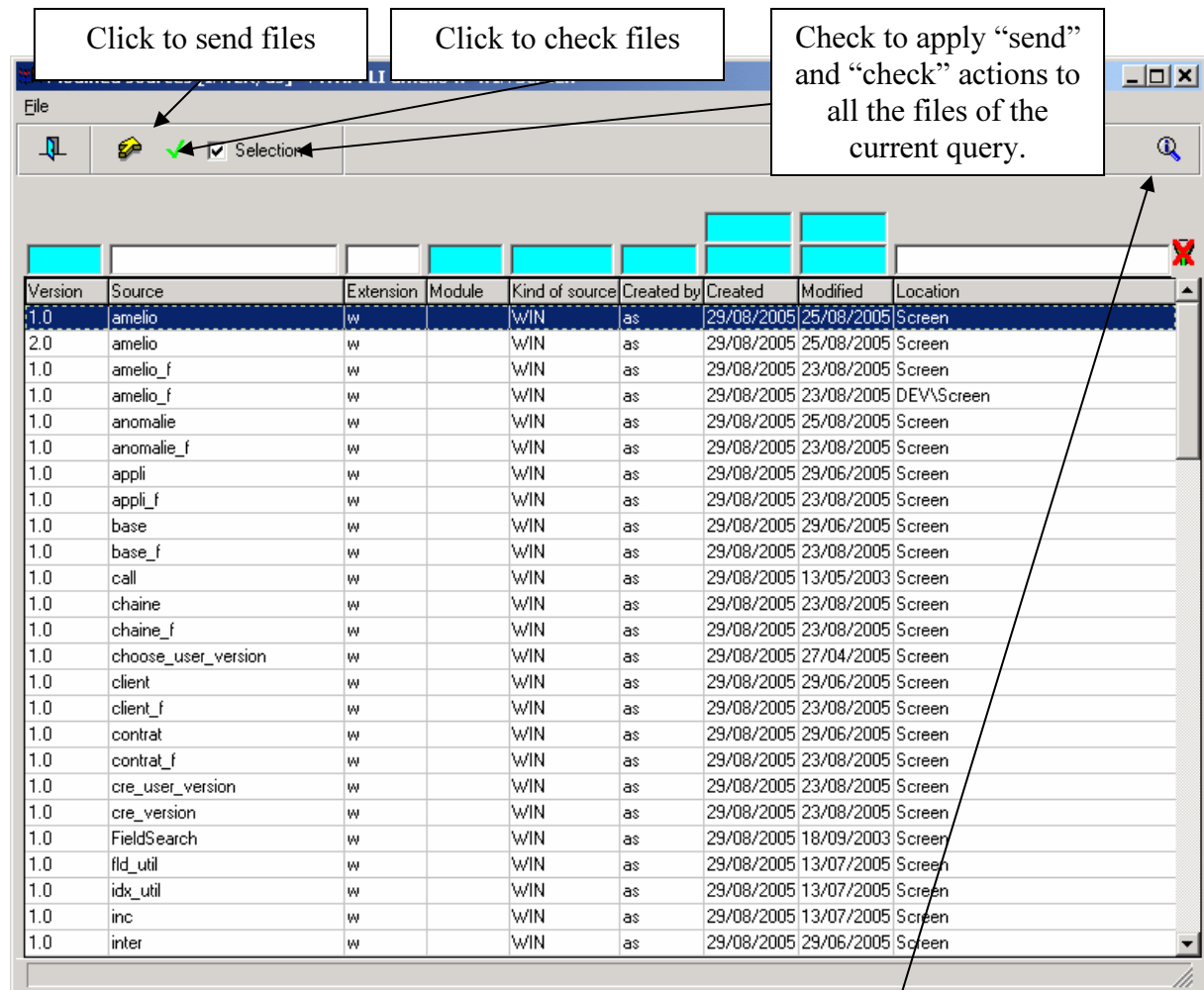
When a file exists in development environment, it is not a very good idea to modify the original version because there is a high risk of inconsistency between the versions of the file. So, if IcePRO lets you do it, it displays a warning when you open such a file in ADE tools.


You can not save a new file in the development directory with ADE tools. A file manually put in development directory is never registered.


e) Sources exchange between sites

IcePRO allows you to synchronize sources between sites. When you are working with ADE tools, each time you save a file, IcePRO flags it as modified. That is, it must be sent to other sites. For files modified with other tools (exe files, pictures, ...), IcePRO can check the file date to detect modification.

You send files to other sites by using *Intervention->Send sources*



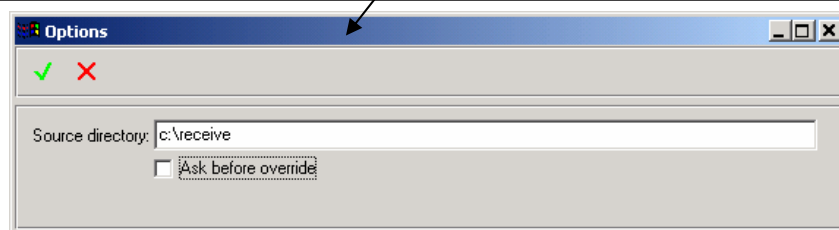
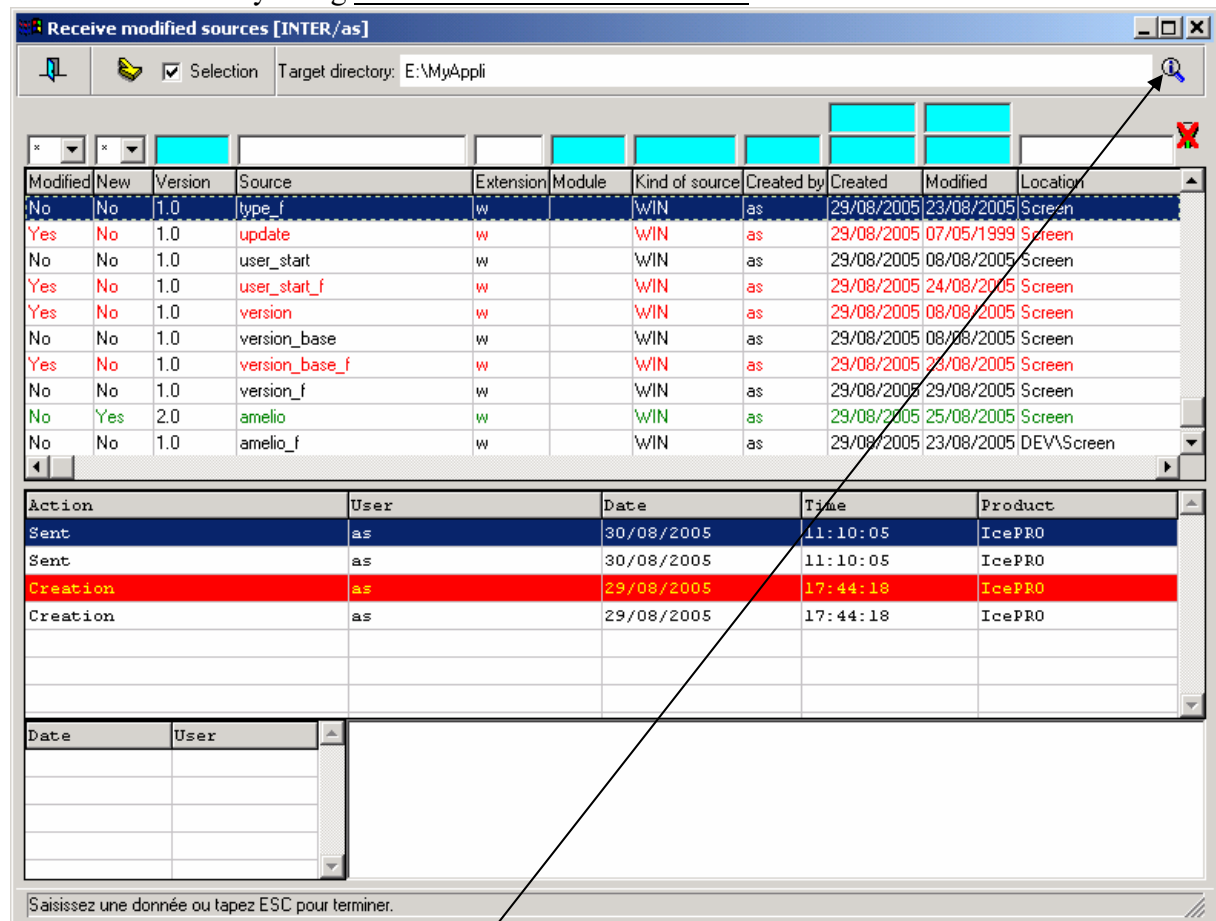
The options screen let you define the target directory. It must be a different one each time you send sources, unless you delete its content after each use. Depending on the types configuration (use file date or not) and state of the box “Search for files modified without ADE at start”, IcePRO begins or not to search modified files when it starts. For Progress 4LG files, it uses internal flag. After that, all the modified files are listed. Use the  button to send files. IcePRO duplicates the application tree to send the files, that is you can get several subdirectories under your target directory. The best is to use a zip tool to convert the whome tree in a single zip file in order to send it easily to other sites.

The  button clears the list of the modified files. Never use it before you are sure you have received all files from other sites. Indeed, when you receive files, IcePRO compare these files to your own modified files to detects conflicts.

The exchange process must follow these steps:

1. All sites send its modified files to each other.
2. All sites run “Receive sources” to check conflicts, and resolve them.
3. All sites run “Receive sources” to get the files.
4. All sites check its own sources.

You receive files by using *Intervention->Receive sources*



When a site receives files from another site (by using zip file sent by email or any other way), it saves it on a local or network drive, then runs this program. The first action is to run the Options screen to type the directory where received files are located. After that, the list of the files to get appears. Following cases are possible:

A file was created on the remote site. It appears in green. It is flagged as new.

A file was modified on the remote site. It appears in black.

A file was modified on both sites. It appears in red. It is flagged modified.

The third case is a conflict. You have to deal with it. There is no particular method to resolve conflict. Conflict just should never appear, but they do. At this moment, the first action to take is to back up your own local file. Then, you can receive file with “Ask before override” option, thus, you can choose to not override a particular file in a conflict. Or you can choose to receive all files then to restore one of the backed up files.



Remember to check all your files (*Intervention->Send sources*) when you have received files from all other sites.

Exchange between sites does not only synchronize files. Sources records and histories are also synchronized. If a new version is created on a site, receiving sources from this site automatically creates the version on your site, as it duplicates all your environment records to use this new version.

f) Application deployment

Before using sources deployment for a version, this version must have at least one customer. (customer can be your own company, internal departments,...).

To add customers, run *File->General->Customers* and insert a record.

To add customer to a version, run *File->Application->Versions*, edit the version by using  button, and add customers by using  button.


Note : When you add a customer to a version, this customer is removed from any previous version.

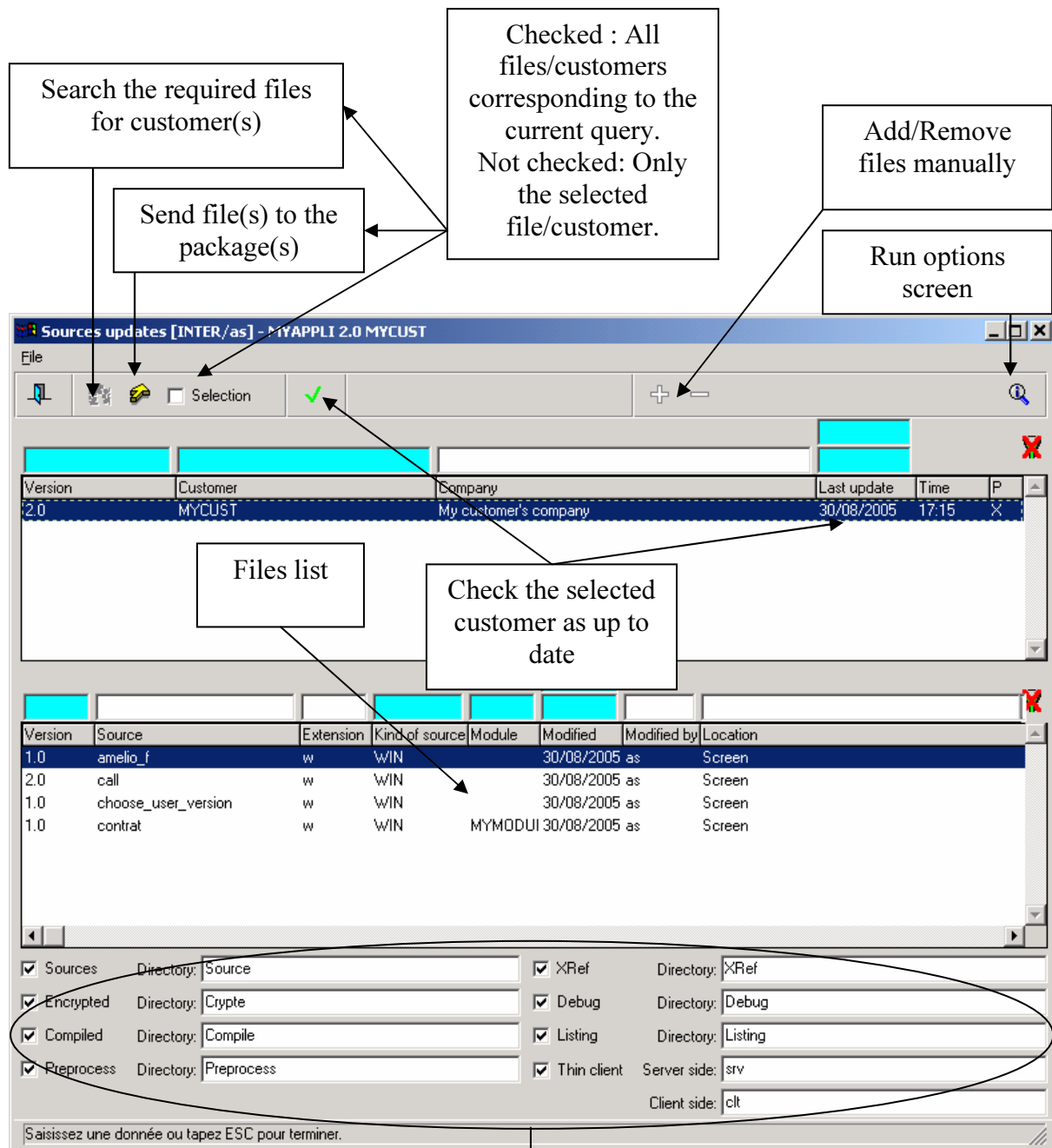
You can generate various types of deployment packages by using *Intervention->Application deployment*

- ⇒ r-code and no compilable files: to be delivered to customers
- ⇒ Preprocessed/encrypted sources: to be compiled on other platforms such as Unix or Linux or to be compiled on customers sites.
- ⇒ Debug, listing or X-Ref compilation for debugging purpose

IcePRO can help you to prepare you packages by searching the files which must be included. The search is based on file modifications, but this not a simple test to detect which files were modified since the last update provided to the customer. IcePRO uses its analysis features in order to give you all required files, modified or not. For instance, modifying a single include file can result in large update because a lot of .p or .w use it. The process also takes care of versions.

You can also add and remove sources manually from a package. By using filtering, sorting and multi records selecting capabilities of IcePRO screens, sources analysis and histories, you can easily know which files you must provide.

Note : Most of IcePRO screens contain filters frequently working as “Begins” filters (some of these are “contains” filters). You can also click on the head of browse columns to sort the records. A first click results in an ascending sort. A second click results in a descending sort. When you add sources manually, you can select several sources by using your mouse and CTRL key. You can select all records of the current query by using  button.



Search the required files for customer(s)

Send file(s) to the package(s)

Checked : All files/customers corresponding to the current query.
Not checked: Only the selected file/customer.

Add/Remove files manually

Run options screen

Files list

Check the selected customer as up to date

Version	Customer	Company	Last update	Time	P
2.0	MYCUST	My customer's company	30/08/2005	17:15	X

Version	Source	Extension	Kind of source	Module	Modified	Modified by	Location
1.0	amelio_f	w	WIN		30/08/2005	as	Screen
2.0	call	w	WIN		30/08/2005	as	Screen
1.0	choose_user_version	w	WIN		30/08/2005	as	Screen
1.0	contrat	w	WIN	MYMODUI	30/08/2005	as	Screen

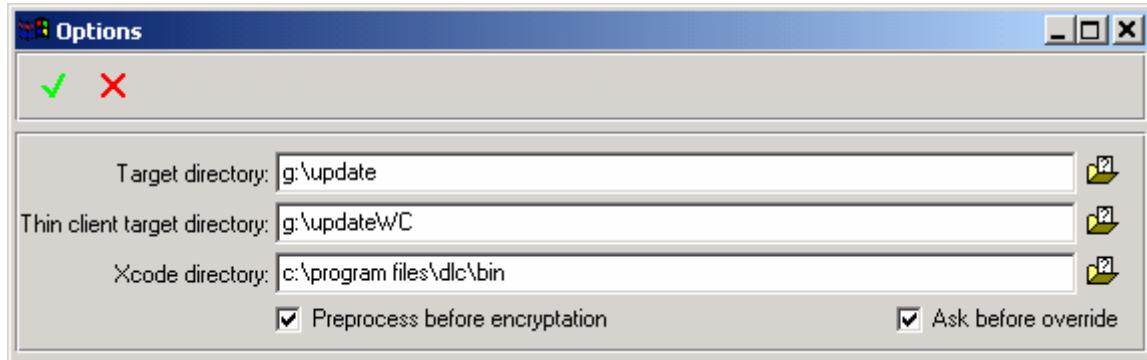
☒ Sources Directory: Source ☒ XRef Directory: XRef
☒ Encrypted Directory: Crypte ☒ Debug Directory: Debug
☒ Compiled Directory: Compile ☒ Listing Directory: Listing
☒ Preprocess Directory: Preprocess ☒ Thin client Server side: srv
 Client side: clt

Saisissez une donnée ou tapez ESC pour terminer.

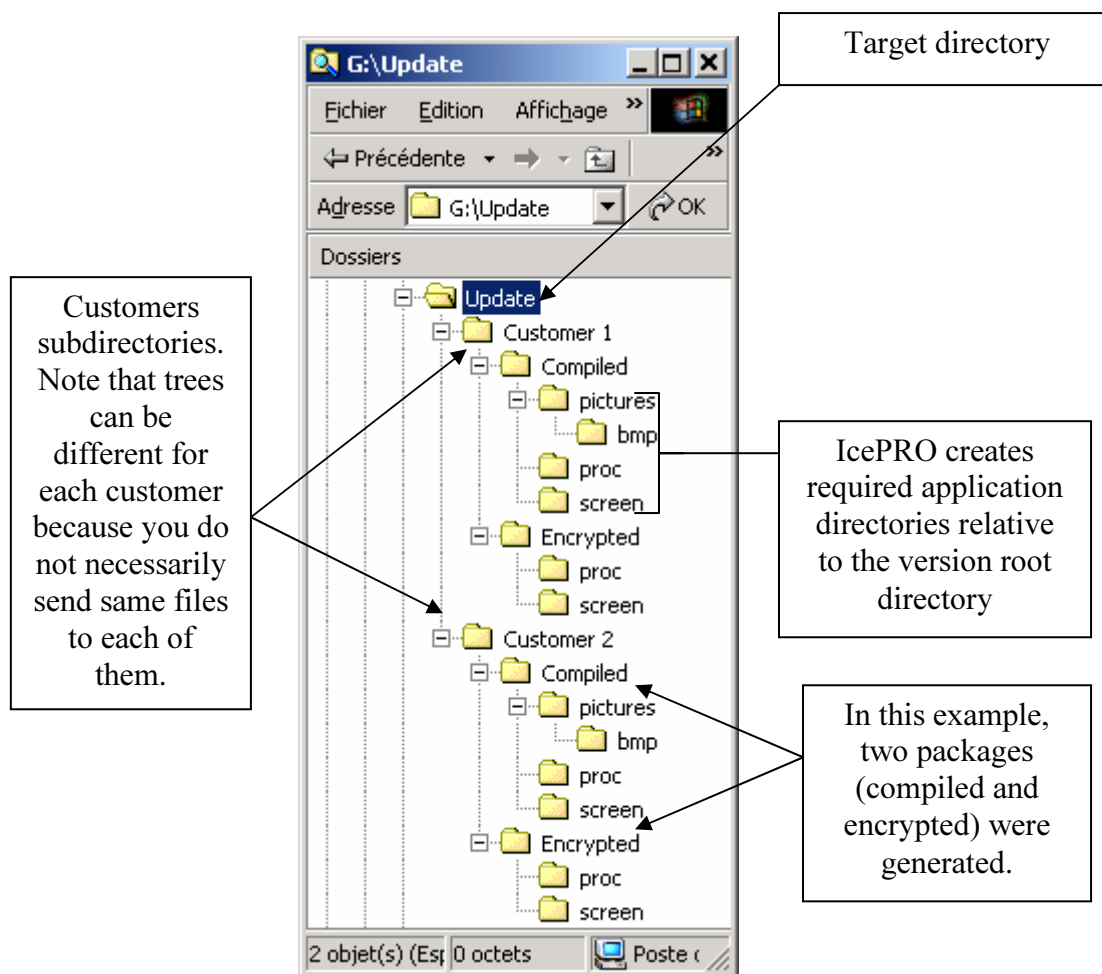
Define which packages to generate, and where to put them. The directories typed here will be subdirectories in the target directory defined in the options screen. Each directory must be different from the others.

Options screen let you define:

- ⇒ Target directory: Where the generated files will be created.
- ⇒ Thin client target directory: Same purpose as the Target directory in a Web Client application deployment context.
- ⇒ Xcode directory: Where to find xcode.exe. Required to generate encrypted files only.



Under the target directory, you get one subdirectory per customer. Under each customer subdirectory, you get one subdirectory per package. Under each package subdirectory, IcePRO put each application source in the required subdirectory:



Lets describe each package:

1. Sources

Files are copied.

2. Compiled

Files are compiled. The files which are not compilable (.exe, .bmp, ...) are just copied.

3. Preprocess

Files are preprocessed, that is all includes references are substituted by their contents and all preprocessors (&SCOPED-DEFINE, &GLOBAL-DEFINED) are replaced by their value. Such files can be compiled as is because includes files are not required anymore.

4. Encrypted

Files are encrypted by using Xcode Progress utility. If you check the box "Preprocess before encryption" in the options screen, sources are preprocessed first. All files which are not compilable are copied.

Encrypted files can be compiled with Progress runtime environments (Client Networking) instead of development environments (Provision, OpenEdge or 4LG development).

5. XRef

File are compiled with -XREF option for debugging purpose. The files which are not compilable are ignored. See Progress documentation for more information about XREF compiler option.

6. Debug

File are compiled with -DEBUG option for debugging purpose. The files which are not compilable are ignored. See Progress documentation for more information about DEBUG compiler option.

7. Listing

File are compiled with -LISTING option for debugging purpose. The files which are not compilable are ignored. See Progress documentation for more information about LISTING compiler option.

8. Thin client

This is a variant of the compiled package. A thin client application using Progress Web Client technology includes two kind of files: client side files (included r-code files, pictures, ...), which stand or run on the client machines and server side files (often only r-code files) which run on a server machine using Progress appserver session. It may be useful to generate these r-code files in separated directories. Thin client package is generated in the same way as compiled package but in separate target directories and sub directories. Not compilable files are also just copied. You use IcePRO source record thin client properties, "Thin client – Client side" and "Thin client – Server side" to tell IcePRO on which side a file must reside. Note that a file can be simultaneously "Server side" and "Client side".

Note : All packages are created according to version requirements as long as you use IcePRO to find which files are required for a customer. Especially, propath can be different depending on the customer version and IcePRO environment used. So, be very careful when you add files manually because IcePRO let you the entire choice concerning these files and you will be able

to add files from any version. Keep in mind that adding “Vn” file version to a customer using “Vn+1” version while the “Vn+1” version of the same file exists, can result in inconsistent package.

Files which are under development environment are never proposed by IcePRO until they are put again in production environment.

*Correctly define source types “compilable” property is very critical.
It is essential to keep analysis up to date. (see “Evaluation” part of this manual)*

*IcePRO considers that your files are free of compilation errors while generating packages.
Prospective compilation errors will be silent and will result in incomplete or absent files.*

The encrypted option includes some limitations due to Progress Xcode utility:

- ⇒ *Xcode utility can not override an existing file. So, before re-generate an encrypted package, be sure to remove all existing encrypted files.*
- ⇒ *Xcode utility can be unable to deal with string parameters longer than 120 characters. Path files resulting in such strings may not be generated.*

Remember to check the customer after your package has been created. Only the files which were included in the package will disappear for the next use.

V. IcePRO-ADE interactions

a) ADE Tools

Using IcePRO modifies the behaviour of:

- ⇒ AppBuilder
- ⇒ Procedure Editor
- ⇒ Procedure Windows
- ⇒ Propath editor

Appbuilder, Procedure Editor and Procedure Windows get a new entry in their menu bar:

- ⇒ IcePRO : Runs the IcePRO main screen/
- ⇒ Informations : Displays IcePRO informations about the current file.
- ⇒ Opened sources : Lists all the opened files. “See” button bring the selected file in foreground.
- ⇒ Execute : To run a Progress executable file.
- ⇒ Compile : Runs IcePRO compiler
- ⇒ Document : Opens or creates a Word document for the current source.
- ⇒ Personal : Run <IcePRO install directory>\PROCED\Adecomm\perso.p. This source comes with IcePRO. You can customize it.
- ⇒ Compile on save (Procedure Editor and Procedure Windows only) : This is a toggle-box. It corresponds to the “Compile on save in editors” property of a source record. See “Compile on save feature” below for more informations.
- ⇒ ** IcePRO – Sinfoni ** [Application name / version]/user: This not a real menu-item because it trigs no action. It is a way to quickly know in which Application/Version you are working.
- ⇒ About IcePRO : About box.

Propath editor: IcePRO runs its own propath editor instead of Progress propath editor. This allows you to use the alias [root] instead of real path.

Note : IcePRO menus and IcePRO Propath editor are added dynamically by using Progress API, without modifying any of the original Progress files. You can always use your Progress products without IcePRO. You can install any Progress service pack over IcePRO.

Always use the [root] alias, never use the literal value of the version root directory.

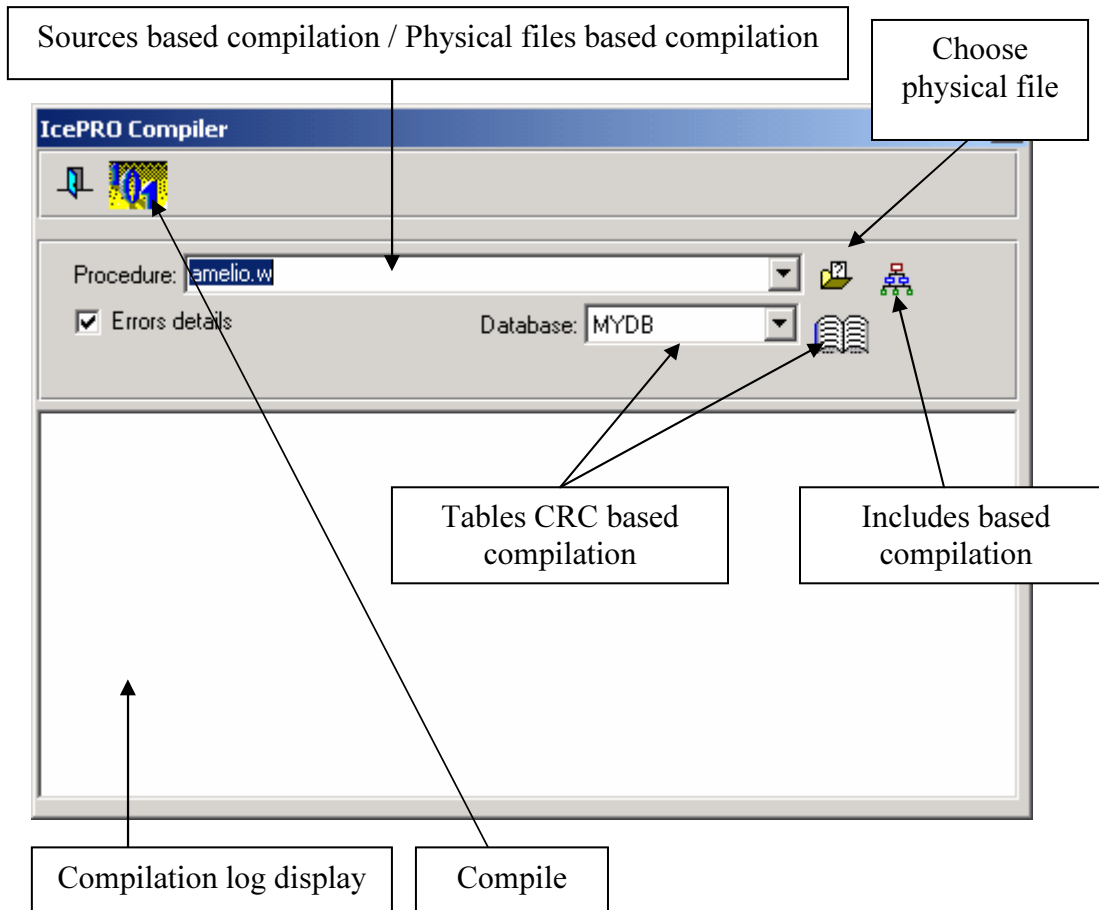
Since propath is dynamically managed by IcePRO, modifying propath in windows registry has no effect in IcePRO environments.

You can not modify propath with IcePRO propath only for the current session. Any modification is saved immediately. This is because IcePRO gives you the possibility to create as many environments as you need, each of these having its own propath. To work with alternative propath, just duplicate an existing environment and modify it according to your needs.

b) IcePRO Compiler

ADE->IcePRO->Compile

Note : You have to correctly define Database(s) (File->General->Database) and connection(s) (File->Application->Connections) before using IcePRO compiler.



In all compilation modes, the r-code files are always saved under the current environment version root directory. For instance, compiling a Vn source in Vn+1 environment put the r-code under the Vn+1 root directory (or subdirectory if needed), and not under Vn root directory. If needed, subdirectories are created on the fly in Vn+1 to reflect Vn tree. The r-codes databases references are recorded in the source record.

- ⇒ Sources based compilation: Type a source name without any reference to its path. In this mode, IcePRO queries all source records of your application corresponding to this name and compile them. The character “*” is allowed everywhere in the source name (several times if needed). IcePRO uses the Progress “MATCHES” syntax to find records. IcePRO implement a version priority mechanism. It searches sources in descending version order, beginning from the current environment version. If a source exists for more than one versions, only the first source found is compiled (that is the highest version). Trying to compile Vn+1 source which does not exist in Vn version while you are working in Vn environment has no effect. These mechanisms allow you to run any version of your application at any time. Note that you can always compile sources by using original Progress compiler, but none of these mechanisms will be implemented, so you have to deal with versions, target directories and propath manually. This can be a very hard task in a multi versions context.
- ⇒ Physical files based compilation: Type a file including its full or relative path (“*” is not allowed). By using full path, you know which source is compiled. By using relative path, the file is searched in the propath. On the contrary to the “Source based compilation”, version priority is not implemented. That is, you can compile a Vn-1 source in a Vn environment, even if the Vn source exists or compile a Vn+1 source in a Vn environment. As the r-code files are however saved in the current environment version directory, be sure of what you are doing when using this compilation.
- ⇒ Includes based compilation: Click and choose one or more include files (by using mouse and CTRL key), then launch the compilation. IcePRO uses its analysis capabilities to retrieve all compilable files which use the include files you have chosen. Thus, you can compile quickly all sources affected by an include modification.
- ⇒ Tables CRC based compilation: Select database, then click and choose one or more tables (by using mouse and CTRL key), then launch the compilation. IcePRO uses its analysis capabilities to retrieve all compilable files which use the tables you have chosen. Thus, you can compile quickly all sources affected by a database schema modification.

Note : Includes and tables CRC based compilation rely on application analysis. To be efficient, the analysis must be up to date.

While compiling, the full pathname of the files are display, so you can know which physical files are compiled.

IcePRO considers all not compilable sources as possible include files. Do not be surprised to see other files than “.i” in the include files list.

c) “Compile on save” feature

A default IcePRO behaviour is to compile source immediately after saving it, even if you are working with the Procedure Editor or a Procedure Window. IcePRO also replaces the AppBuilder “compile on save” default mechanism, which compiles source in the same directory. As the IcePRO compiler, IcePRO “Compile on save” feature always saves r-code under the current environment version root directory.

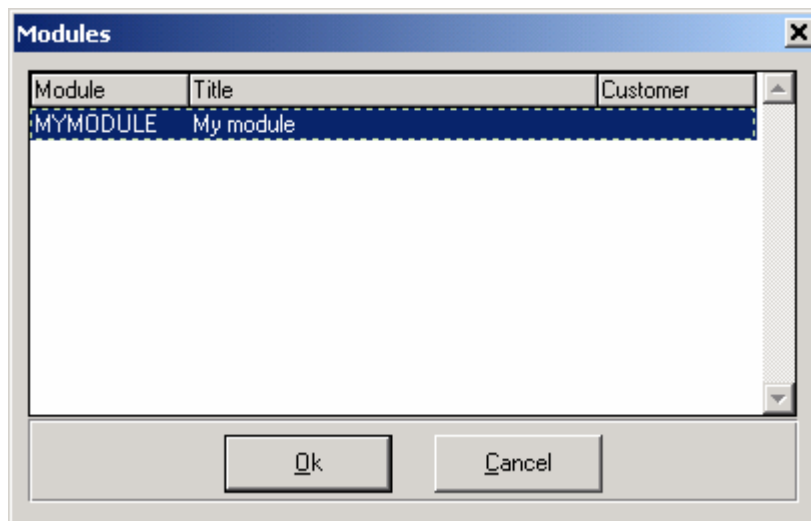
Let's name S the source version and E, the environment version. You save the source, what can happen ?

- ⇒ $S > E$: IcePRO does not compile.
- ⇒ $S = E$: IcePRO compiles.
- ⇒ $S < E$ and the same source exists in version S' with $E \leq S' < S$: IcePRO does not compile.
- ⇒ $S < E$ and there is no source in version S' with $E \leq S' < S$: IcePRO compiles.

This reproduces the version priority mechanism of the IcePRO compiler.

d) Saving files

The first time you save a file with ADE tools under IcePRO environment, it is registered. IcePRO displays a screen where you have to choose the application module to which the source belongs (to create modules, runs *File->Application->Modules*).



IcePRO registers the source only if you save it under one of the version root directories. A source record is created in IcePRO database. The source version is given by this directory. A first level record history is created where creation date, hour and user are recorded.

When you save a registered source, a first level history record is created where modification date, hour and user are stored. This history record is created when you close your file. Thus, opening a source, saving it several times, then closing it, creates only one history record. A new history record will be created if you open it again, save it again and close it again. Closing a registered source after saving it also creates a second level history record where you can describe your modifications, change the module to which the source belongs, and its “thin client” properties, that is “Server – side”, “Client –side”, both or none. When you close your file, IcePRO displays this screen:

Describe your modifications - [MYAPPLI]

✓

Source: MyProc Extension: p

Location: proc

Module: MYMODULE

☐ Thin client - Client side

☐ Thin client - Server side

IcePRO creates only one second level history record per date and per user.



Note : Progress Procedure Windows and Procedure Editor are text editors. You can use them to create any text files (.bat, .ini, ...), not only 4LG sources. This way, you can use history features.

When you open a file then close it without having saved it, a first level history record "Consultation" is created where date, hour and user are stored.

You can not save a file in development directory.

e) Moving or renaming files

You move or rename a file when you use "Save as.." menu item.

- ⇒ Rename a file: IcePRO ask you if you want to delete the original file. This prevents you to leave unused files in your directories.
- ⇒ Move a file from a directory to another under the same version root directory: IcePRO ask you if you want to delete the original file. This prevents you to leave unused files in your directories.
- ⇒ Move a file from a directory to the same directory under another version root directory: IcePRO considers that you send a source to another version. You can also send a file to a higher version by using  button in File->Application->Sources.
- ⇒ Move a file to a development directory: This action is not allowed since you can not create files in development directory. To do it, use  button in File->Application->Sources.

All of these actions create the corresponding first level history record.

f) Automatic file backup

IcePRO creates a backup of each file which can be overridden or deleted in the following way:

- ⇒ You save a file by using ADE tools.
- ⇒ You override a file by using ADE tools.
- ⇒ You delete a file after using “Save as...” to move or rename it.
- ⇒ You receive files from another site.
- ⇒ You delete a file by using Sources screen.
- ⇒ You retrieve a source from the development environment.


The backup directory is **C:\IcePROSrcBkp**, that is backup are located on the local system. Under this directory, IcePRO creates one directory per application, then one directory per version, then reproduces your application tree. The naming convention is Year1.month1.day1-Year2.month2.day2-number-name.ext where Year1, month1, day1 represents the current date, Year2, month2, day2 represents the OS file date, number is the internal source number, and name.ext is the original file name. This allows you to get one backup per day and per computer as an existing backup file is never overridden. Remember this feature. It can help you to save precious time in case of file loss.

g) First level history records

This is the list of history records you can find:

- ⇒ Consultation : Open and close file without saving it.
- ⇒ Modification : Open save and close a file.
- ⇒ Creation : Register a source (save, search, analysis)
- ⇒ Replace : Override a file by using save menu.
- ⇒ Put in development mode
- ⇒ Move : Move a file by using “Save as...” menu and delete the original file.
- ⇒ Analysis : IcePRO analysis.
- ⇒ Retrieved from development mode
- ⇒ Sent : Send file to other sites.
- ⇒ Received : Receive file from another site
- ⇒ Sent to higher version
- ⇒ Received from lower version
- ⇒ Sent to lower version
- ⇒ Received from higher version
- ⇒ Generation : Create a file by using IcePRO source code generator.

You can consult sources histories first and second levels by using



File->Application->Sources. Use  button to see source details and select History folder.


You can search for first level histories by using Intervention->Sources histories.


You can search for second level histories by using Intervention->Sources modifications. In this screen, the filter corresponding to the “Description” column use **CONTAINS** syntax, that is you can use “*” character to replace end of words, and any combination of words, !(or operator), & (and operator) and brackets.

h) Sources screen


File->Application->Sources

You can open a Progress source file or any text file in a Procedure Window by using  button. The check box “Display absents” displays records for which there is no existing physical file in gray. Use the  button to delete these records.

You can open or create Word documents by using  button.

You can search database references in r-code files by using  (apply to current filters), and use the “DB references” and “DB required” filters.

Note : The “DB References” filter works as a “CONTAINS” filter. Retrieved r-code informations are r-code file, CRC, languages, database references, and codepage. These informations are also retrieved when you compile sources by using IcePRO compiler or “Compile on save” feature.

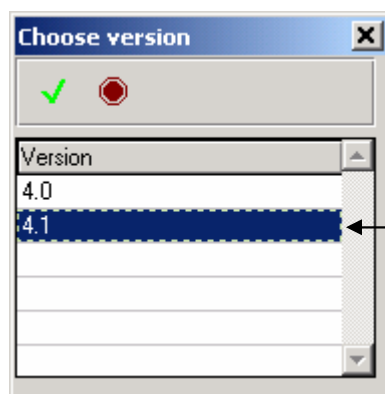
You can analyze a progress source by using . (see next part of this manual).

You can modify consistency control by using .


i) Consistency control

As a source can exist under several versions, consistency control helps you to open the right source versions under the right environment versions.


- ⇒ When you open a physical file by using File->Open in ADE tools, Icepro searches all the existing versions of the file, and let you choose the right one. It does not matter that you choose the source in the first version directory or in subsequent version subdirectories. If several versions exist, this screen appears:

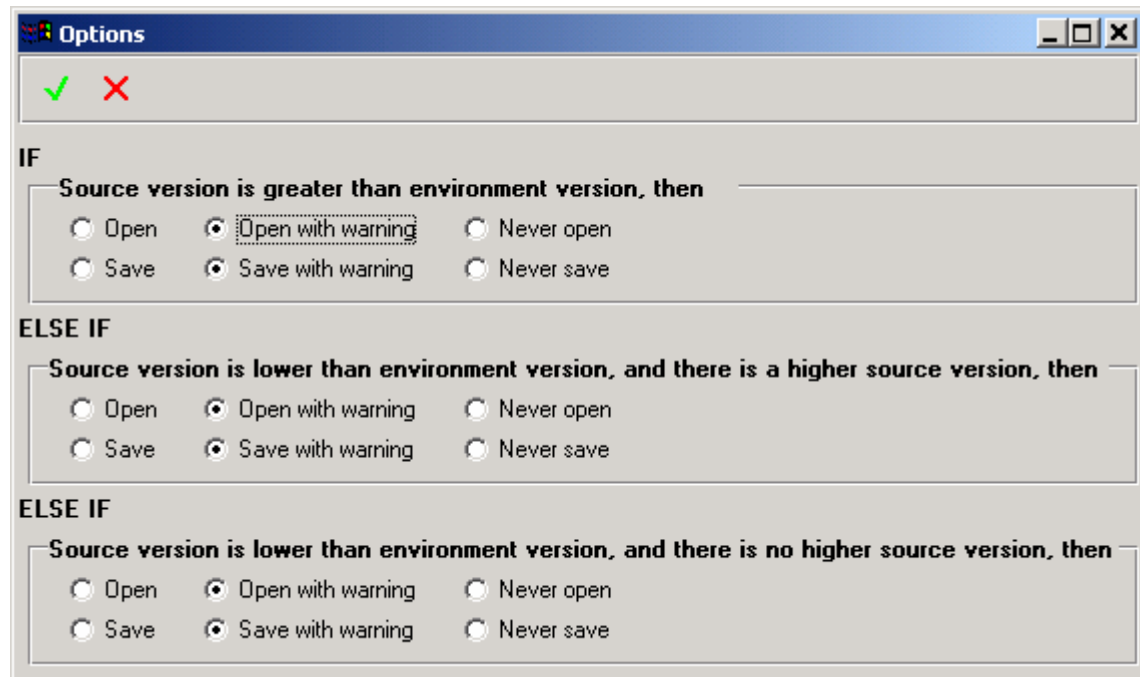


IcePRO selects the most logical version according to your current environment and version priority mechanism.

Select the version you want to open, and click  (or double click or press **Return** key).

- ⇒ When you open a source by using File->Open in ADE tools or by using “open” feature in IcePRO screens, IcePRO controls the consistency between the source version and you current environment version. You can customize how IcePRO handles inconsistencies. For instance, if you open/save a V2.0 source under a V1.0 environment, IcePRO can ask you for confirmation, prevents you to open/save the file

or process your request without any warning. You configure this behaviour by using  in the sources screen.




Open/Save with warning is the default. That is IcePRO ask you for confirmation before opening/saving the file.

VI. Evaluation

a) Source analysis

Evaluation->Sources analysis

Note : You can also analyze a progress source by using  in File->Application->Sources.

This is one of the most useful features. Analysis allows you to get the following informations for each compilable source:

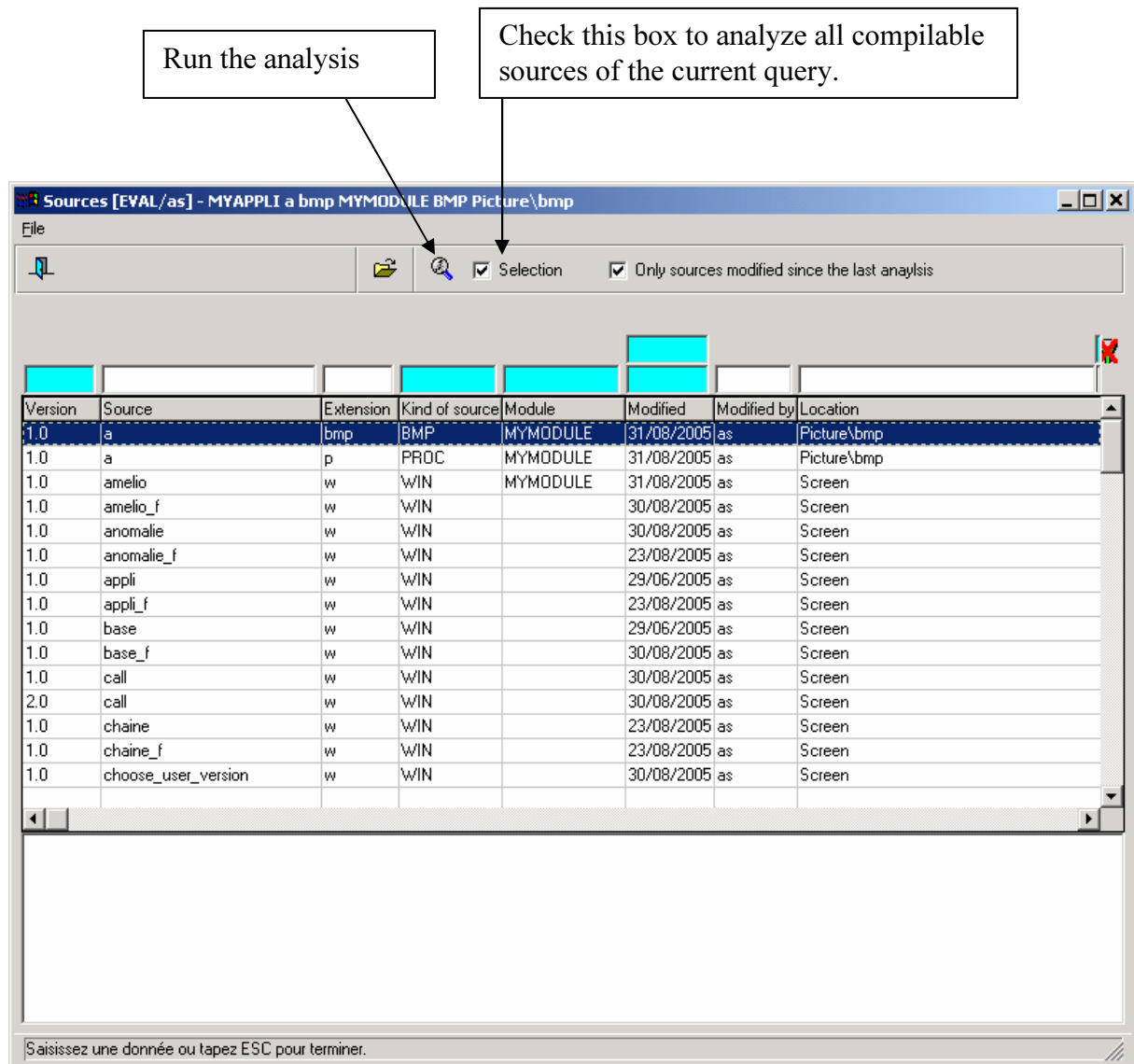
Which includes are used ?



Which calls are done to other sources ?

Which tables are used ? How ? (that is access, update, delete, create, search or reference)

Which fields are used ? How ? (that is access, update, or reference)

Which indexes are used ? Is it in an efficient way ?



You should run daily analysis (at least each day you make modifications) in order to keep consistent informations (for instance as a night process). As it can take a while, run it on an efficient computer. The analyzer runs as a multi tasks process, that is you can launch an analysis, iconize the screen and continue to work with your environment in the same Progress session. While analyzing,  button is replaced by the  button. Click on it to stop the analysis.

Analysis depends on the version environment you are running. Indeed, a source can use different includes and indexes under different environments because propath and databases are different. So, if you have several versions of an application, you have to run an analysis for each one. Analysis uses the “Version priority” rules. Sources which do not match these rules are ignored (for instance, sources in version V_{n+1} are not analyzed under a V_n environment).

The box “Only sources modified since the last analysis” means sources which were modified or using includes which were modified.

You can see a summary of the informations retrieved while analysing by using *File->Application->Sources*. Complete informations are available in *Evaluation* menus as described below.

b) Uses

⇒ Table uses

Evaluation->Tables uses

Database	Table	Source	Extension	Modifié le	Location
harmonic	DEPOT	s_pf_mod	p	02/09/2005	proced
harmonic	ETAT	s_pf_mod	p	02/09/2005	proced
harmonic	FOURNI	s_pf_mod	p	02/09/2005	proced
harmonic	F_CDE	s_pf_mod	p	02/09/2005	proced
harmonic	F_CDE_L	s_pf_mod	p	02/09/2005	proced
harmonic	F_PF	s_pf_mod	p	02/09/2005	proced
harmonic	MODELE	s_pf_mod	p	02/09/2005	proced
harmonic	NOMEN	s_pf_mod	p	02/09/2005	proced
harmonic	PF	s_pf_mod	p	02/09/2005	proced
harmonic	PF_STOCK	s_pf_mod	p	02/09/2005	proced

Location	Source	Extension	Line	Action
proced\inc	sto_edit	i	49	REFERENCE
proced\inc	sto_edit	i	216	SEARCH
proced\inc	sto_edit	i	216	ACCESS
proced\inc	sto_edit	i	217	ACCESS

Header: Database, table and analyzed sources

Lines: Kind of use, physical sources and line number.

A “table use” is a set of records, header and lines. The header record identifies which analyzed source is using which table. A table is usually used several times in the same source. Furthermore, the use can be made through an include file. The lines display all the uses, and in which sources they are written. In the example above, you can see that the source *s_pf_mod.p* uses the table “DEPOT” from “HARMONIC” database through the *sto_edit.i* include. That is that the source code is not written in *s_pf_mod.p* but in *sto_edit.i*. To see that source code, just right click on the line record and choose *Open* menu. The file will be opened in a procedure window with the cursor positioned on the correct line number.

The kinds of use are ACCESS, SEARCH, UPDATE, DELETE, CREATE, REFERENCE.

ACCESS : A record is read.

DELETE : A record is deleted.

UPDATE : A record is updated.

CREATE : A record is inserted.

REFERENCE : Reference to the table without read/write.

⇒ Field uses

Evaluation->Field uses

This is the same feature as Table uses, but at field level. The kinds of uses are REFERENCE, UPDATE and ACCESS.

⇒ Index uses

Evaluation->Index uses

Index uses [EVAL/as] - HARMONIC4 4.1 HARMONIC9 F_PF F_PF_PK s_pf_mod p proced

Database	Table	Index	Source	Extension	Index mistake	Location
HARMONIC9	FOURNI	FOURNI_PK	s_pf_mod	p	No	proced
HARMONIC9	F_CDE	F_CDE_ETAT	s_pf_mod	p	No	proced
HARMONIC9	F_CDE_L	PF	s_pf_mod	p	No	proced
HARMONIC9	F_PF	F_PF_PK	s_pf_mod	p	No	proced
HARMONIC9	MODELE	MODELE_PK	s_pf_mod	p	No	proced
HARMONIC9	NOMEN	NOMEN_FK2	s_pf_mod	p	No	proced
HARMONIC9	PF	PF_FK5	s_pf_mod	p	No	proced
HARMONIC9	PF_STOCK	PF_DEP_DLC_PICK	s_pf_mod	p	No	proced
HARMONIC9	PF_STOCK	PF_STOCK_PK	s_pf_mod	p	No	proced
_TEMP-TABLE	tt_appsrv	default	s_pf_mod	p	No	proced
_TEMP-TABLE	tt_f_cde	PK	s_pf_mod	p	No	proced
_TEMP-TABLE	tt_val	PK	s_pf_mod	p	No	proced

Index	Location	Source	Extension	Line	Mistake
F_PF_PK	proced\vinc	sto_edit	i	767	No
F_PF_SK1					
F_PF_SK2					

Field
K_FOURNI
K_PF

Right click to see the 4GL source.

Fields in the selected index

The index used is selected by default. You can look at other ones to verify the relevance of the used index.

If "yes", this indicates a possible bug or non relevant index use.

In addition to header and line records, the screen displays all the indexes of the current table, and the fields used in each one. Temp-table indexes are not displayed.

If a `FIND` statement uses ROWID, the primary index will be displayed as the used index. If a statement uses RECID, you will find the word "RECID" instead of an index name.

Index mistake flag indicates a possible bug or non relevant index use. Usually, this is because the fields used as criteria are not indexed. In this case, Progress uses the primary index. It could also be caused by a bug in 4GL statement. For instance, your application uses a table ORDER and a table ORDER_L. They contain respectively orders and orders lines.

ORDER.ORDER-NUM is the order number.

ORDER_L.ORDER-NUM is the order number in the lines table. If you write :

```
FIND ORDER WHERE ORDER.ORDER-NUM = "050900001" NO-LOCK NO-ERROR.
FOR EACH ORDER_L WHERE ORDER.ORDER-NUM = ORDER.ORDER-NUM EXCLUSIVE-LOCK:

    DELETE ORDER_L.
END.
```

Instead of

```
FIND ORDER WHERE ORDER.ORDER-NUM = "050900001" NO-LOCK NO-ERROR.
FOR EACH ORDER_L WHERE ORDER_L.ORDER-NUM = ORDER.ORDER-NUM EXCLUSIVE-LOCK:

    DELETE ORDER_L.
END.
```

then Progress compiles it without any message. However, in the first case, the statement will delete all the order lines in the database... because the criteria `ORDER.ORDER-NUM = ORDER.ORDER-NUM` returns the TRUE value.

IcePRO analysis helps you to prevent such a mistake.

⇒ Include uses

Evaluation->Include uses

Displays all include files.

⇒ Source uses

Evaluation->Source uses

Displays all calls.

Note : IcePRO analysis is a compile time analysis. Dynamic queries, dynamic buffers and dynamic calls (that is `RUN VALUE(<file-name>)`.) are not handled. These cases can be handled in a different way by the "Session analysis" which is a runtime analysis.

IcePRO relies on Progress XRef compilation. This introduces some limitations:

The line number can be an approximation and some temp-table statements are omitted. These are Progress Xref compilation issues.

When IcePRO records a table, index or field use through an include, an include use or a source use, it registers this source or include file if it is not already registered. However, it is possible that your application uses files that can not be registered because they are not under the version root directory. In this case, IcePRO records the full pathname of the file instead of the relative pathname.

The database names used are those you have created in Files->General->Databases, not the logical names. The reserved database name “_TEMP-TABLE” is used for the temp-tables.

c) Unused objects

Unused objects shows the tables, index, includes and sources which were not detected as used during the IcePRO analysis.

These screens make a search in real time when you launch them. This can take a while with databases containing a lot of tables and index, and applications with a lot of files.

Dynamic queries, dynamic buffers and dynamic calls are not detected during the compile time analysis, so there is no guaranty that an unused object is really unused. For instance, IcePRO considers all non compilable files as potential includes, so all non progress files will be considered as unused includes. If you never run analysis, all objects will be “unused”.

On the contrary, if an object is not listed here, you can be sure that it is used somewhere.

Informations always concern the environment version you are using. An unused object in version N can be used in version N+1.

⇒ Unused tables and index

The screenshot shows the 'Unused files and index [EVAL/as]' window. It contains two main panes. The top pane, labeled 'Unused tables', displays a list of tables from the HARMONIC9 database. The bottom pane, labeled 'Unused index', displays a list of indexes from the same database. Both panes have a search bar at the top and a list of results below. The 'Unused tables' pane shows tables like ACTION, ACTION_PROSPECT, AFFAIRE, AGIO, AG_COM, AG_COND, AG_STAT, ALEA, ANA, and APE. The 'Unused index' pane shows indexes like AGENT_FK1, AGENT_WQ, CLIENT_EDI, CLIENT_FK1, CLIENT_FK10, CLIENT_FK11, CLIENT_FK12, CLIENT_FK13, and CLIENT_FK2. A third pane at the bottom shows a list of indexes and a field K_DEVISE.

Database	Table	Name
HARMONIC9	ACTION	?
HARMONIC9	ACTION_PROSPECT	?
HARMONIC9	AFFAIRE	?
HARMONIC9	AGIO	?
HARMONIC9	AG_COM	commission representant
HARMONIC9	AG_COND	Condition commissionnement agen
HARMONIC9	AG_STAT	Stat agent
HARMONIC9	ALEA	?
HARMONIC9	ANA	Sections analytiques
HARMONIC9	APE	?

Database	Table	Index
HARMONIC9	AGENT	AGENT_FK1
HARMONIC9	AGENT	AGENT_WQ
HARMONIC9	CLIENT	CLIENT_EDI
HARMONIC9	CLIENT	CLIENT_FK1
HARMONIC9	CLIENT	CLIENT_FK10
HARMONIC9	CLIENT	CLIENT_FK11
HARMONIC9	CLIENT	CLIENT_FK12
HARMONIC9	CLIENT	CLIENT_FK13
HARMONIC9	CLIENT	CLIENT_FK2

Index	Field
CLIENT_EDI	K_DEVISE
CLIENT_FK1	
CLIENT_FK10	
CLIENT_FK11	

Unused tables

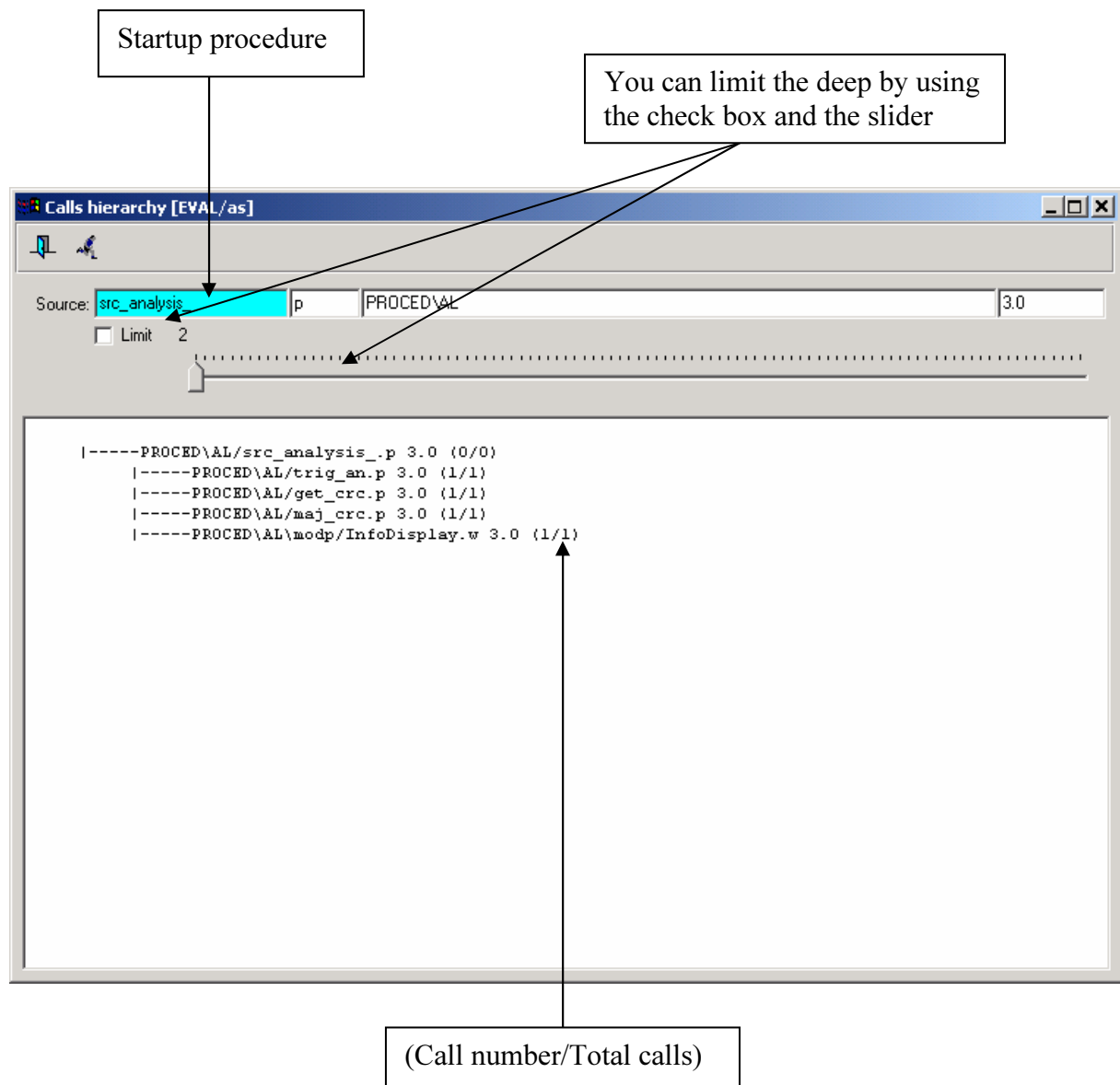
Unused index

⇒ Unused includes
Shows all unused non compilable files.

⇒ Unused sources
Shows all unused compilable files.

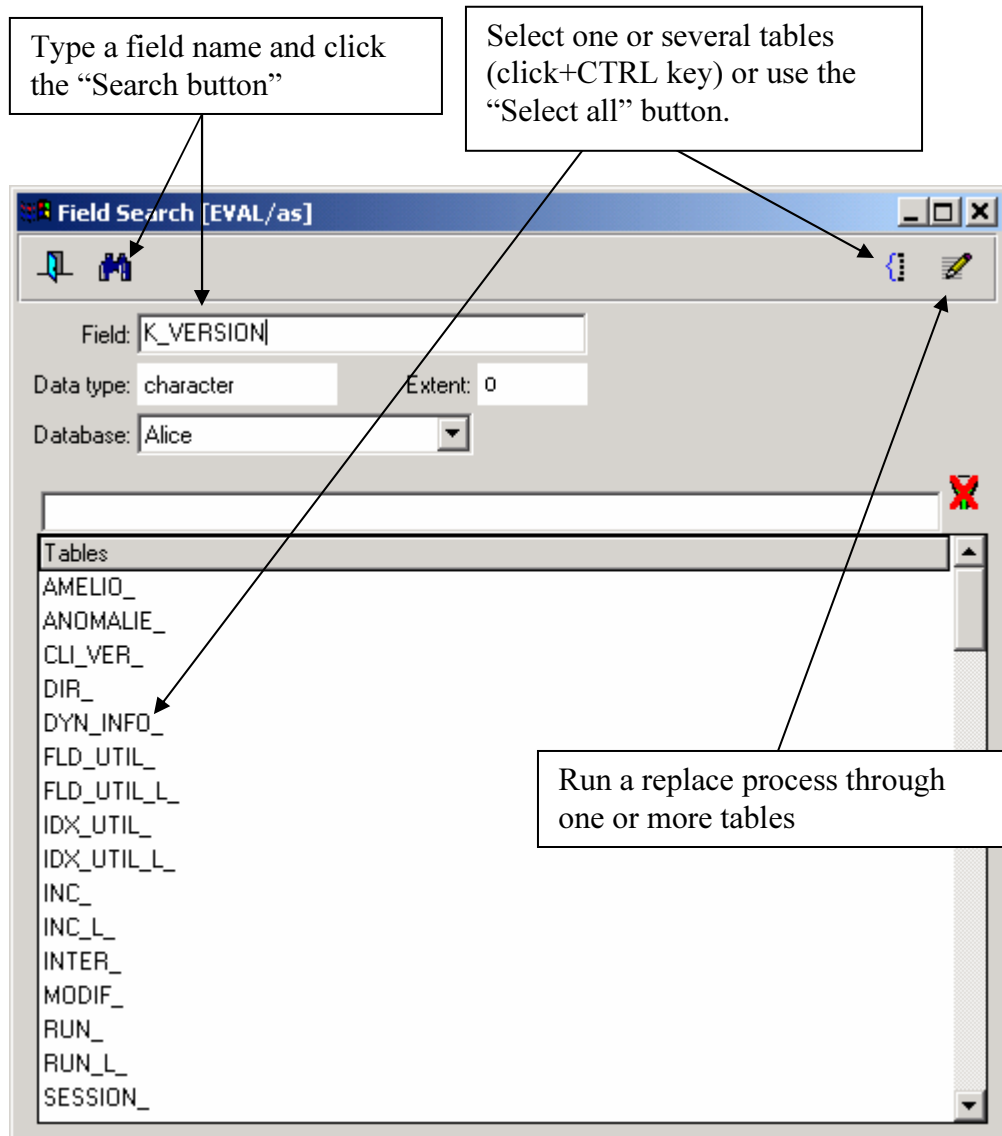
d) Call hierarchy

Displays a call tree of the current version based on the analysis. Keep in mind that dynamic calls (`RUN VALUE (file-name)`) are not handled.




e) Field search

This is a screen where you can search in which tables a field is defined. You can also replace its content in one or several tables. The search is not based on IcePRO analysis. IcePRO just searches the field through the database metaschema (_File and _Field tables).

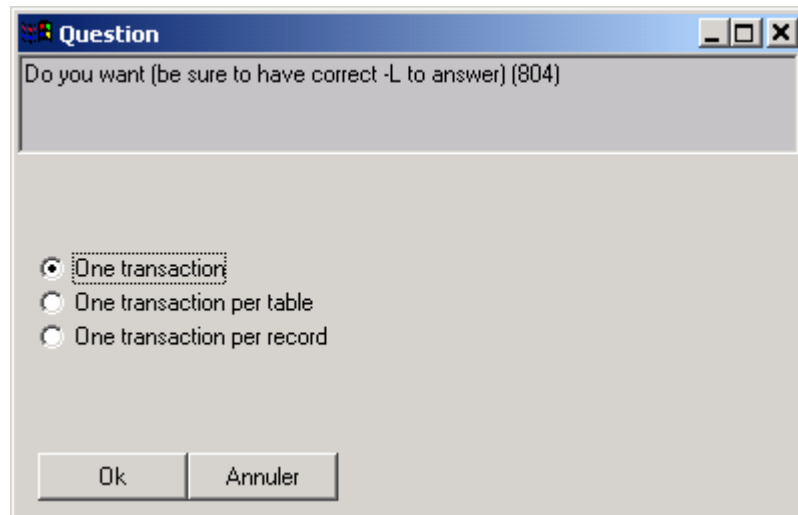


Be careful ! Using the replace process can modify a lot of records in a lot of tables. Be sure of what you are doing before using this feature.

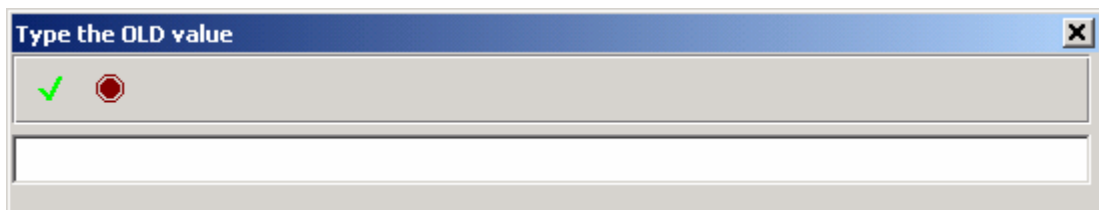
Select one or several tables.

Click on the  button

A message box asks you to choose the transaction mode. This point is very important. You can cancel at this point.

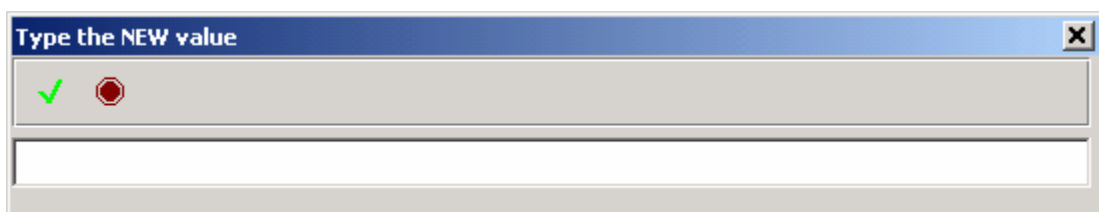


Type the old value of the field. This value will be used to query the records in each table.



Depending on the field data type, the screen may be different. Warning ! Using cancel button do not cancel the process but set the old value to the empty string, null value, or unknown value.

Type the new value you want to assign to the field.



Depending on the field data type, the screen may be different. Warning ! Using cancel button do not cancel the process but set the new value to the empty string, null value or unknown value.

A screen displays the update process, and how many records were updated for each table.

Note : This tool should not be used on a production database, unless you have a reliable backup. The process can be very long, especially if the field you are modifying is not indexed,

because it is used as a criteria. Depending on the transaction mode chosen, you may need to have a high $-L$ parameter set.

f) Version statistics

Evaluation->Statistics


This screen summarizes the analysis informations for the current environment version. The most important field is the percentage of right index uses. It gives you the global performance of your application. However, keep in mind that dynamic queries are not handled.

g) Session analysis

Evaluation->Session analysis

The IcePRO session analyzer is a profiler. It analyzes your programs at runtime. Each executed statement is timed with a precision of a microsecond. The time the programs wait for user inputs is ignored.

To use the IcePRO session analyzer, the application startup procedure must be defined. As IcePRO runs a new progress session, be sure that this new session will be able to connect to the required databases, that is a server is running for each one (IcePRO database is not concerned).

To run a session analysis, create a session analysis record by using  button, type a title and valid a first time. By default, the box "Listings" is checked. This way, IcePRO creates a listing source for each program ran, allowing you to get time for each executed statement. Without listing sources, analysis run faster and need less space disk, but you get only time for each block program. The time spent to create the listing sources is ignored. After the first validation the session parameters which will be used appear in the "Session parameters" editor. These parameters are based on the current session parameters. You can modify all of them but not the "-p" value. The session analysis propath will be the same as the current IcePRO environment. Valid a second time. IcePRO runs a new Progress session by using the application startup procedure.

When the new session is running, use all the programs you want to be analyzed. IcePRO will wait until this session ends, that is it will not respond until the analyzed session executes a **QUIT** statement. Be sure to have such a statement in your application, or you may loose control on your IcePRO session.

When the analyzed session ends, the IcePRO analyzer begins to collate data after a delay which can be in seconds or minutes, depending on the number of programs you have ran and their complexity. Be patient.

Then, in the "General" folder, you can see the list of all blocks or programs which were called, total time, average time of each call, total number of calls, percentage of total session time, percentage per call, average overhead and the corresponding registered sources (or a "?" for unregistered sources). As you can sort data by clicking on every column header (one click = ascending sort, a second click = a descending sort), you can now find the blocks or programs which have spent the longest time during the session. Time is displayed in seconds, with 6 decimals (microseconds).

The screenshot shows the 'Sessions analysis [EVAL/as] - 6802 Match' window. Annotations include:

- Session parameters:** Points to the 'Details' tab and the session information fields (Session number: 6802, Date: 09/06/2005, Version: 4.0, Total time: 4.400140, Title: Match).
- Deep, height and time are parameters used to display the call tree.** Points to the 'Call tree' tab and the 'Deep: 0', 'Height: 0', and 'Time (s): 10' controls.
- The folder "Details" shows informations about the source corresponding to selected block or program.** Points to the 'Details' tab.
- Check it to get time for each statement.** Points to the 'Listings' checkbox.

The 'Details' tab displays session parameters and a list of blocks/programs with their performance metrics.

Block/Programm	Total time	Time per call	Calls	% session	percentage	Average Overhead	Source	Ext
Session	0.000000	0.000000	1	0.0000	0.0000	0.000000	?	?
f:\progress9\alice\proced\val\asSessionAnali	0.005347	0.005347	1	0.1215	0.1215	0.000097	?	?
f:\progress9\harmonic9\proced\login.w	0.029215	0.029215	1	0.6640	0.6640	0.023676	Login	w
api\Windows.p	0.000091	0.000091	1	0.0021	0.0021	0.000086	Windows	p
api\Winfunc.p	0.000068	0.000068	1	0.0015	0.0015	0.000063	Winfunc	p
enable_UI f:\progress9\harmonic9\proced\logi	0.005240	0.005240	1	0.1191	0.1191	0.000032	Login	w
maj f:\progress9\harmonic9\proced\login.w	0.000475	0.000475	1	0.0108	0.0108	0.000023	Login	w
modp/maj.p	0.000614	0.000614	1	0.0140	0.0140	0.000483	maj	p

To see details about a source, select a block or program, then click on the "Details" folder.

Here are displayed only the blocks which belong to the source selected in the folder "General". By default, the block you have selected in "General" folder is also selected here.

Selecting a block displays three kinds of data about it :
Time spent for each line.
Programs/blocks caller
Programs/blocks called

The screenshot shows the 'Sessions analysis [EVAL/as] - 7268 aaa' window. It has tabs for 'General', 'Details', 'Call tree', and 'Memo'. The 'Call tree' tab is active, showing a table of blocks and their call statistics. Below this, there are three sub-tables: 'Line/time', 'Is called by these blocks/programms', and 'Calls these blocks/programms'. The 'Line/time' table lists lines of code with their total and per-call times. The 'Is called by' table shows callers, and the 'Calls' table shows callees. A code editor at the bottom displays the source code for the selected block.

Block/Programm	Total time	Time per call	Calls	% of Session	% per call	Average Overhead
IsAppAlreadyRunning f:\progress9\harmonic9\proced\login	0.000238	0.000238	1	0.0094	0.0094	0.000108
ClientLeger f:\progress9\harmonic9\proced\login.w	0.000371	0.000371	1	0.0147	0.0147	0.000069
ok f:\progress9\harmonic9\proced\login.w	0.003509	0.003509	1	0.1389	0.1389	0.000123

Line	Total time	Time per call	Calls
0	0.000108	0.000108	1
683	0.000048	0.000048	1
685	0.000042	0.000042	1
684	0.000012	0.000006	2
682	0.000010	0.000010	1

Is called by these blocks/programms		
Calls	% Session	Block/Programm
1	0.0613	une_instance f:\progress9\ham

Calls these blocks/programms		
Calls	% Session	Block/Programm
1	0.0051	CreateMutexA api/Windows.p
1	0.0018	WaitForSingleObject api/Winc


```

Si le fichier existe, appel annulé */
DEF VAR vi_mutex AS INTEGER NO-UNDO INITIAL 0.
FUNCTION IsAppAlreadyRunning RETURN LOGICAL (p-OnePerSystem AS LOGICAL, p-AppName AS CHAR):
DEF VAR vi_ret AS INTEGER NO-UNDO.
DEF VAR vc_mutex AS CHAR NO-UNDO.

vc_mutex = p-AppName + ' is running'.
RUN CreateMutexA in gh_api (0.0,vc_mutex, OUTPUT vi_mutex).
  
```

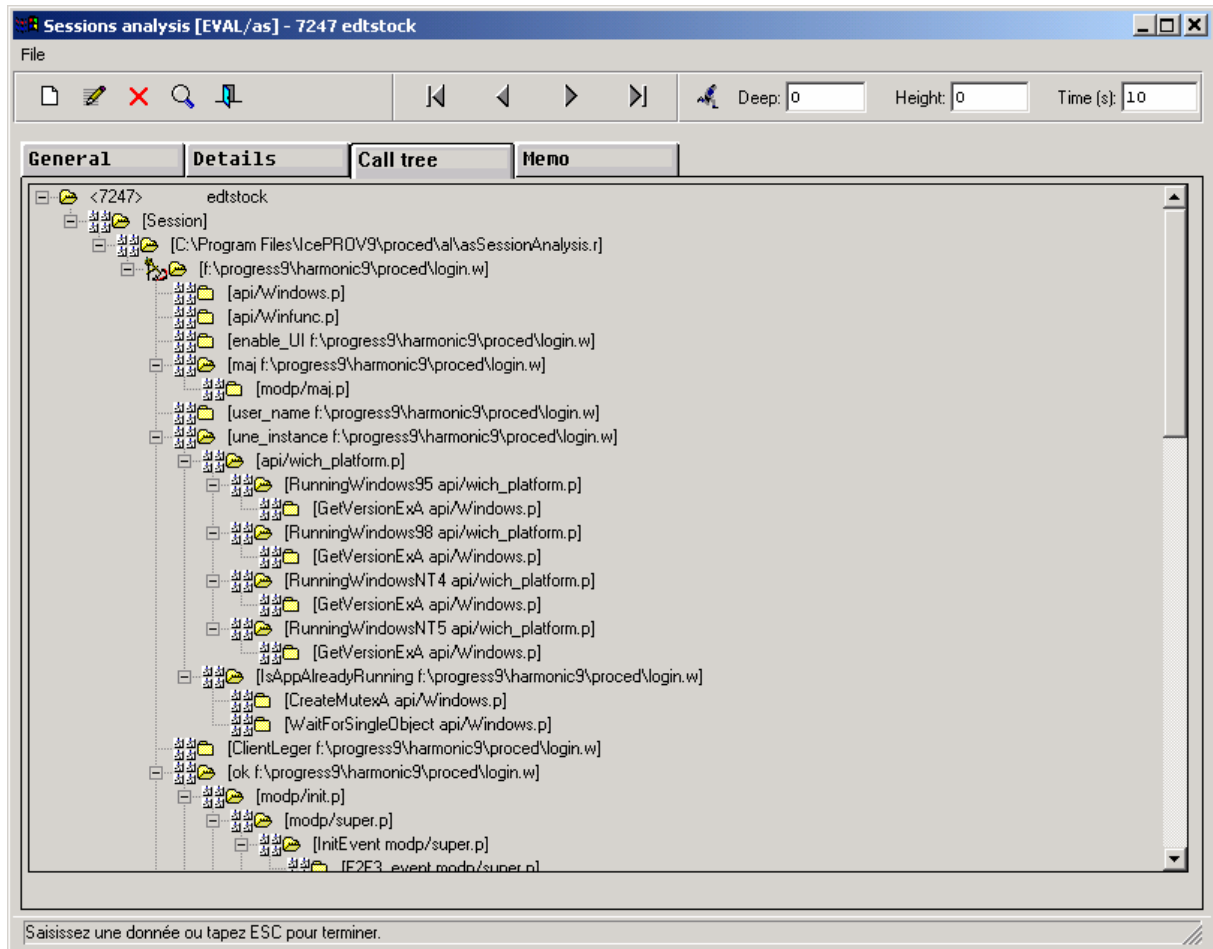
Time spent for each line:
This is the list of the lines which were executed. When you select a line, the source is displayed in the editor. The lines are sorted in descending order. This way, the first line shown is the line which spent the most time in the current block.

Shows the caller(s).
Double clicking on a caller block has the same effect as selecting this block in the "General" folder.

Shows the called blocks.
Double clicking on a called block has the same effect as selecting this block in the "General" folder.

Double clicking on these browses allows you to "walk" in the call tree.


You can see a graphical view of the call tree in the folder “Call tree”.



Deep: Limits the deep of the call tree. “0” means there is no limit.


Height: Limits the number of lines displayed. “0” means there is no limit.

Time: Limits the time allowed to create the tree.


You can modify the default values then use the  button to apply them.


A call to a registered “.w” is symbolized by the Progress Appbuilder icon.


A call to a registered “.p” is symbolized by the Progress Procedure Editor icon.


A call to a user interface trigger is symbolized by .

A call to an internal procedure is symbolized by .

A recursive call is symbolized by .

A end of tree caused by the deep or eight limit is symbolized by .

A end of tree caused by the time limit is symbolized by .

All other cases (unregistered sources, unknown extensions...) are symbolized by .

VII. Creation

IcePRO source code generator was written before the Progress Version 9 was released. Yet, dynamic queries and buffers did not exist. Keep in mind that all you can do with IcePRO source code generator can probably be done by using dynamic queries and dynamic buffers in a better way. Unless you have to build V8, V7 or V6 Progress applications, you should use dynamic code. However, if you find something which can not be done with dynamic code, let us know...

a) Skeleton

File->Skeletons

Source code generation is based on skeletons, that is, text files written in Progress 4GL where some parts of the code were replaced by the IcePRO generation language. This language contains five kinds of words.

- ⇒ Data: <BASE>, <FILE>, <FIELD>. Data are replaced by their value.
- ⇒ Instructions: <BEGINB>, <BEGIN>, <BEGINF>, <END>, <AFIELD>... Instructions allow you to repeat code blocks.
- ⇒ Properties: <FILE-NUM>, <FIELD-TYPE>, <FIELD-VIEW-AS>, <FIELD-FORMAT>... Properties are related to database objects or to the source code itself.
- ⇒ Constants : <USERID>, <TODAY>, <TIME>, <SRC>, <LIBRARY>... There are predefined constants. You can also define your own local or global constants.
- ⇒ Functions: /#f_caps()#/ , /#f_length()#/ ... Functions are stored in libraries. A library is a Progress 4GL source where you write your functions.

Example:

```
DEF TEMP-TABLE tt_count NO-UNDO
    FIELD TABLE-NAME AS CHAR FORMAT "x(30)"
    FIELD NUM-RECORD AS INT FORMAT ">>>>>>>9"
    INDEX PK IS PRIMARY UNIQUE TABLE-NAME.

<BEGIN>
    /* Counts the records in table <FILE> */
    CREATE tt_count.
    ASSIGN tt_count.TABLE-NAME = "<FILE>".
    SELECT COUNT(*) INTO tt_count.NUM-RECORD FROM <FILE>.

<END>
```

When you run IcePRO source code generator, you have to choose a skeleton, databases, tables and fields. Suppose you choose the tables SUPPLIER and CUSTOMER. You will get this code:

```
DEF TEMP-TABLE tt_count NO-UNDO
    FIELD TABLE-NAME AS CHAR FORMAT "x(30)"
    FIELD NUM-RECORD AS INT FORMAT ">>>>>>>9"
    INDEX PK IS PRIMARY UNIQUE TABLE-NAME.

    /* Counts the records in table SUPPLIER */
    CREATE tt_count.
    ASSIGN tt_count.TABLE-NAME = "SUPPLIER".
    SELECT COUNT(*) INTO tt_count.NUM-RECORD FROM SUPPLIER.

    /* Counts the records in table CUSTOMER */
    CREATE tt_count.
    ASSIGN tt_count.TABLE-NAME = "CUSTOMER".
    SELECT COUNT(*) INTO tt_count.NUM-RECORD FROM CUSTOMER.
```


The best way to learn about source code generation language is to look at the “CHECK” sample skeleton.

All skeleton files and libraries have to be saved in `<IcePRO install directory>\PROCED\AL\ske` directory. Skeleton file must have a “.ske” extension. You write or modify skeleton files in Progress Procedure Editor or Procedure Windows. In order to use a skeleton, create a skeleton record in the IcePRO database.

The skeleton code must match the skeleton file name. If the skeleton file is `<file-name>.ske`, the code must be `<file-name>`.

Overview of the file. You can not modify the file here.

To open the skeleton file or the library

File

General Memo

Skeleton: CHECK Test

Library: check.p Name: Check1 P

Destination: tests

```

*****
*
*   BEGIN[,n[,m]], BEGINF[,n[,m]], BEGINB[,n[,m]]
*
*
*   Begin a code repetition on the databases(BEGINB),
*   tables (BEGINF) or field(BEGIN).
*
*   Two possible uses:
*   <BEGIN[F|B]> [code to repeat] <END>
*
*   you'll get:
*   [code to repeat] [code to repeat] [code to repeat] ...
*
*   <BEGIN[F|B]>
*   [code to repeat]
*   <END>
*
*****

```

Saisissez une donnée ou tapez ESC pour terminer.

Library (optional).

Prefix. (optional).

Suffix. (dot and extension).

Target directory relative to the root version directory.

File name can be any combination of literal strings and `[<BASE[n[,m]]> [<FILE[n[,m]]> [<FIELD[n[,m]]>]`

b) Constants

There are three kinds of constants:

⇒ Global build-in constants

These are predefined constants which you can use in every skeletons.

<USERID>: IcePRO database current user

<TODAY>: Current date

<TIME>: Current time.

<LOC>, <SRC>, <EXT> : Location, source and extension of the file you are generating.

<APPLI>: Application.

<VERSION>: Version.

<SQE>: Skeleton name.

<LIBRARY>: Library name.

⇒ Global constants

Globals constants are those you define by using Creation->Constant. It can be a multi lines text.

⇒ Local constants


You can declare local constants in the skeleton file:

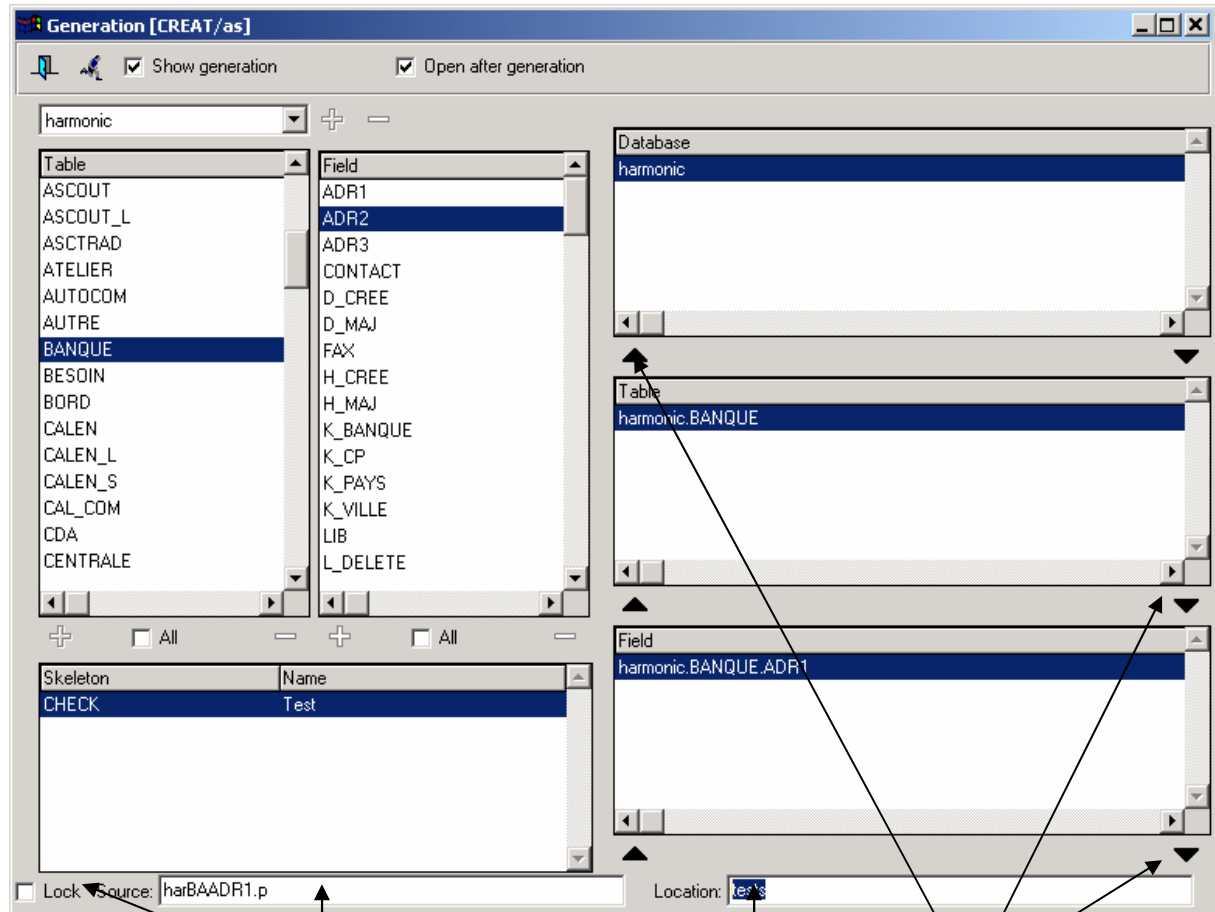
<CONST> <CONT-NAME> Hello world !

Each time IcePRO source generator will find <CONST-NAME>, it will replace it by the string "Hello world !". You can override any global constant by defining a local one.

c) Source generation

Creation->Source generation

This screen allows you to add databases, tables and fields names by using  buttons.



Name of the file which will be generated. The name is based on skeleton data and databases/tables/fields currently chosen. Check the box "Lock" to type another name.

Sort databases/tables/fields.

Target directory

d) Functions libraries

You can write functions libraries. A library is a Progress 4GL program saved under the <IncePRO-Install-Dir>\Proced\AI\Ske directory. It must contain **CHARACTER** functions receiving one **CHARACTER** parameter. For instance:

```
FUNCTION f_caps RETURNS CHAR (pi_param AS CHAR):
    RETURN CAPS(pi_param).
END.
```

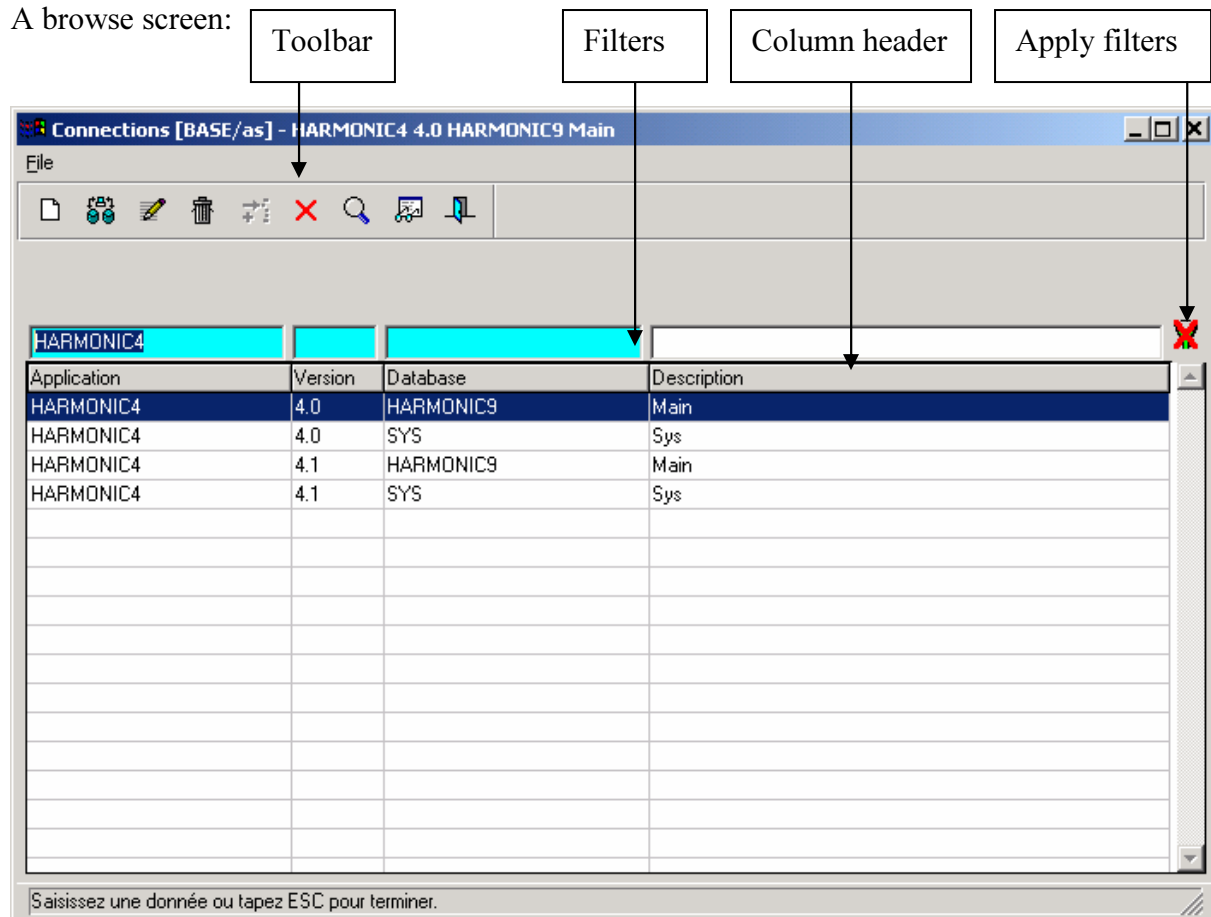
Any occurrence of `"/#f_caps(hello world !)#/"` will be replaced by `"HELLO WORLD !"`

VIII. User interface

a) Screens

Some screens let you create, update, and delete records in IcePRO database while some others are designed for special IcePRO purposes. The first ones are always couples of two screens : The browse which is a screen with a browse object displaying several records, and the form which displays one record details.

A browse screen:



⇒ Toolbar

Buttons are “Create”, “Copy”, “Edit”, “Logical delete”, “Logical recover”, “Delete”, “Search”, “Details”, “Exit”. “Create” and “Copy” both create records, but “Copy” uses the current record as a template. “Logical delete” do not delete records. They are just marked as deleted. Logical recover undoes a “Logical delete”. “Search” allows you to use the form screen to search a record. It can be useful if you want to search records by using criteria different from the filters. Instead of filling filters, the form screen appears where you can use every enabled field as a filter (operator “=”). Each modified field will be used as a filter. To use the blank value, just type any character, then remove it. As the field was modified, it will be used, even if it contains blank value.

⇒ Filters

Filters let you make queries. Most of character filters use **BEGINS** operator while some of them use **CONTAINS** operator. Numeric filters use \geq operator. Logical filters use $=$ operator. Most of date filters are intervals and use \leq and \geq operators. Some filters are blue. These are

foreign keys. You can double click or use the **F5** key on any blue field to get a new screen where you can choose among all possible values. When the focus is not on a filter, **F8** puts the cursor in the first filter. When the focus is on a filter, **F8** applies current filters. **CTRL-F8** discards any filter and any sorting. **F9** shows you informations about the current query : Query text, indexes used, records count. Note that the records count is accurate only if you go to the last record by using the **End** key before the **F9** key.

*Note : **F9** and sort feature work if the browse has filters. If a screen contains several browses, **F9** shows informations about all the browses.*

⇒ Column header

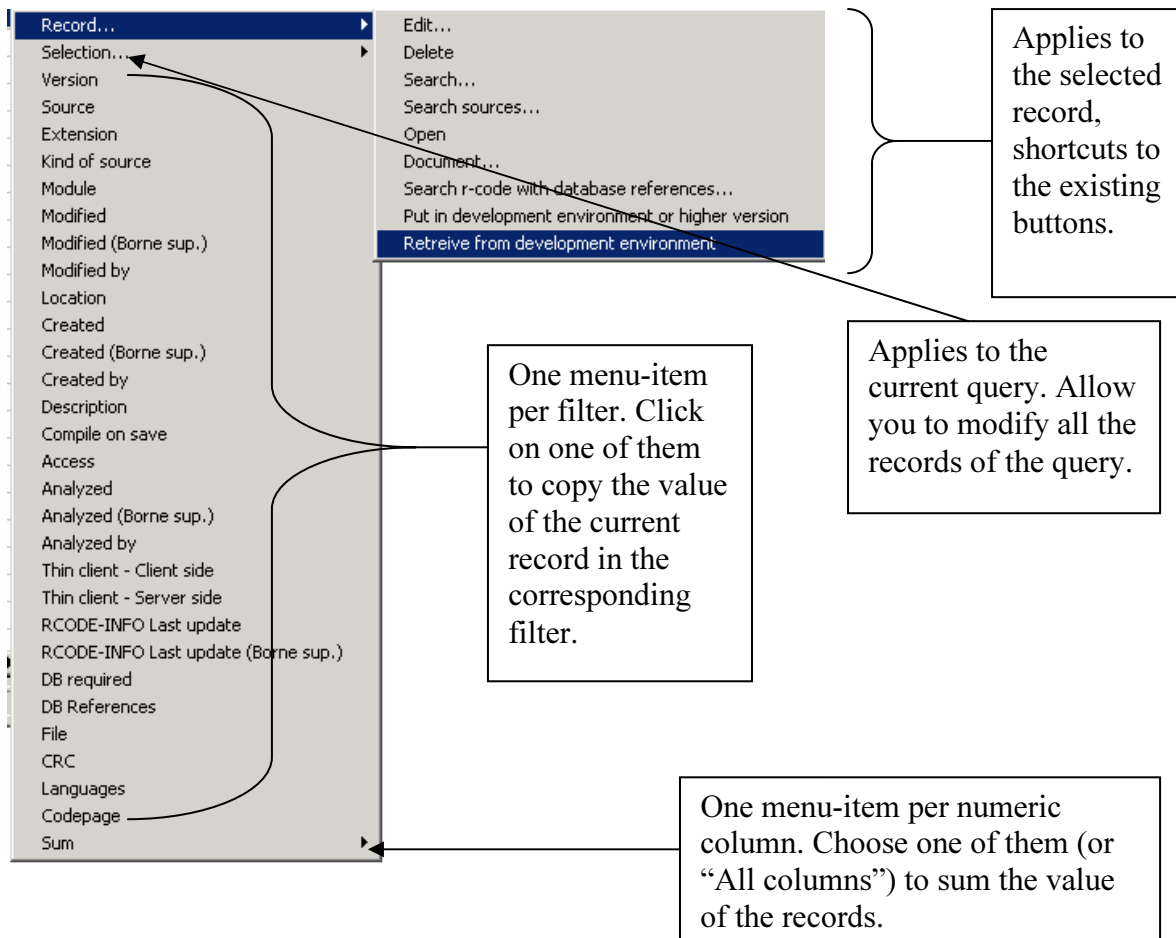
Column headers allow you to sort the records. One click for an ascending sort. A second click for a descending sort.

⇒ Apply filter

This button applies the current filters or discards them if they are already applied (= **CTRL-F8**).

⇒ Popup menu

A right click on the main browse drops a popup menu like this.



A form screen:

Connections [BASE/as] - HARMONIC4 4.0 HARMONIC9 Main

File





Database: HARMONIC9 Harmonic pour version 9

Application: HARMONIC4 HARMONIC PROGRESS V9

Version: 4.0

Description: Main

To create a record, follow these steps:

1. Click the  or  button. The primary key fields are enabled. Fill in these fields.
2. Click the  button to validate the primary key fields. The modifiable fields are enabled. Fill in these fields.
3. Click the  to validate the record.

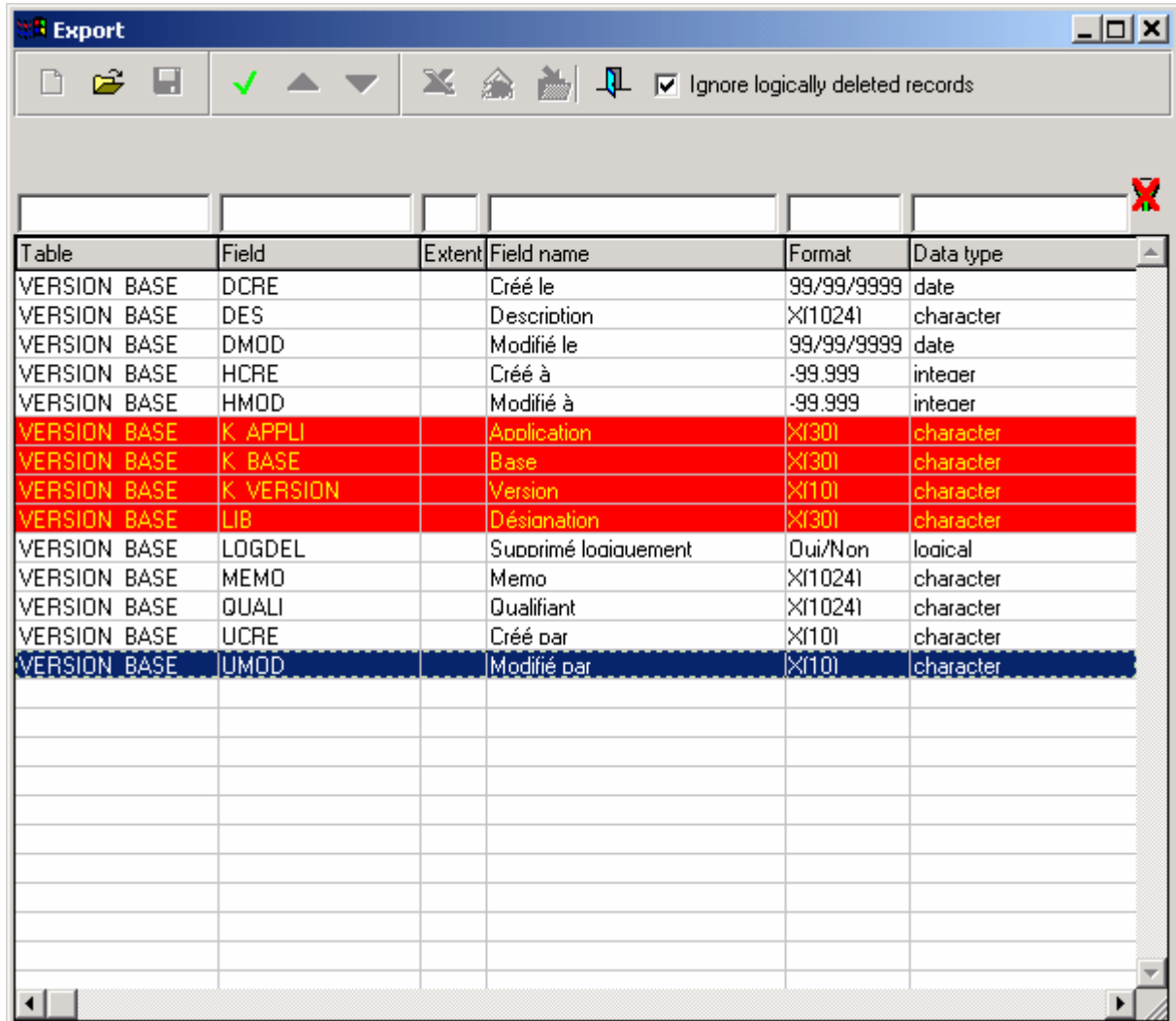
You can double click or use the **F5** key on any blue field to get a new screen where you can choose among all possible values.

Note : The character "Space" is not allowed in primary key fields. These fields are convert to upper case.

The **F6** key shows informations about the users who have created/modified the records.

b) Exporting to Excel

All records of a query in a browse with filters can be exported to excel, to a CSV file or to an ASCII file. Just select the browse, and use **F7** key.



1. Select which fields you want to export by double clicking on them. A selected field appears in a different colour. To select all the fields, do not select at all.

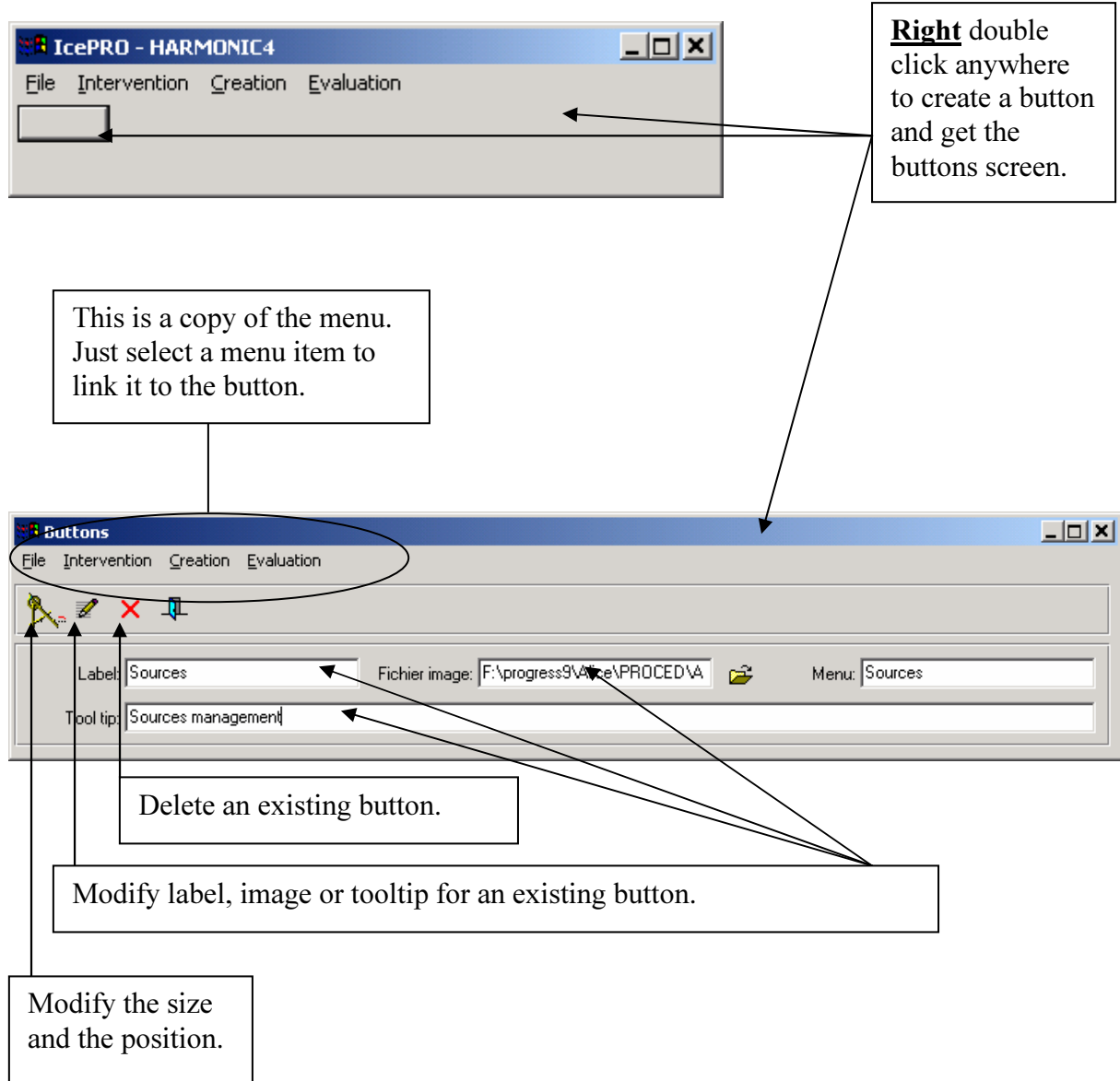
2. Click on . Only selected fields will be kept. For each field, you will get a data column.

3. Use and to move fields up and down. You can also modify the values in the 'Field name' column.

4. Click the or to send records to Excel, CSV file or ASCII file.

c) Creating buttons

In the IcePRO main menu, each user can create its own shortcut buttons to the menu items.



1. Right double click anywhere in the IcePRO main screen.
2. Choose a menu item in the copy of IcePRO main menu.
3. Always modify the position of the button. Do not let it in the left corner. Modify its size if needed. When you have finished with size and position, left click anywhere on IcePRO main screen to get focus on buttons screen. Exit this screen.
4. To modify or delete an existing button, just right click on it.

d) Login

You can use `-U userid [-P password]` parameters in the IcePRO.pf file to provide the userid to IcePRO. This way, the login screen does not appear, and there is no need for the user to type its userid/password.

See Progress documentation for more informations about these parameters.